

Spinnaker C

2.0.0.0

Generated by Doxygen 1.8.17



<b>1 Module Index</b>	<b>1</b>
1.1 Modules	1
<b>2 Data Structure Index</b>	<b>3</b>
2.1 Data Structures	3
<b>3 File Index</b>	<b>5</b>
3.1 File List	5
<b>4 Module Documentation</b>	<b>7</b>
4.1 Spinnaker C Definitions	7
4.2 Camera Enumerations	8
4.3 Chunk Data Structures	9
4.4 Spinnaker C QuickSpin API	10
4.4.1 Detailed Description	10
4.5 QuickSpin Access	11
4.6 Spinnaker C API	12
4.6.1 Detailed Description	12
4.7 Error Handling	13
4.8 System Access	14
4.9 InterfaceList Access	15
4.10 CameraList Access	16
4.11 Interface Access	17
4.12 Camera Access	18
4.13 Image Access	19
4.14 Event Access	20
4.15 ImageStatistics Access	21
4.16 Logging Event Data Access	22
4.17 Device Event Data Access	23
4.18 Chunk data access	24
4.19 Spinnaker C Handles	25
4.20 Spinnaker C Function Signatures	26
4.21 Spinnaker C Enumerations	27
4.22 Spinnaker C Structures	28
4.23 Spinnaker C GenICam API	29
4.24 Node Map Access	30
4.25 Node Access	31
4.26 IValue Access	32
4.27 String Access	33
4.28 IInteger Access	34
4.29 IFloat Access	35
4.30 IEnumeration Access	36
4.31 IEnumEntry Access	37

4.32 IBoolean Access	38
4.33 ICommand Access	39
4.34 ICategory Access	40
4.35 IRegister Access	41
4.36 Spinnaker C GenICam Handles	42
4.37 Spinnaker C GenICam Enumerations	43
4.38 SpinVideo Recording Access	44
4.39 Transport Layer Enumerations	45
4.40 TLDevice Structures	46
4.41 TLInterface Structures	47
4.42 TLStream Structures	48
4.43 TLSystem Structures	49
<b>5 Data Structure Documentation</b>	<b>51</b>
5.1 _actionCommandResult Struct Reference	51
5.1.1 Detailed Description	51
5.1.2 Field Documentation	51
5.1.2.1 DeviceAddress	51
5.1.2.2 Status	51
5.2 _quickSpin Struct Reference	52
5.2.1 Field Documentation	64
5.2.1.1 AasRoiEnable	64
5.2.1.2 AasRoiHeight	64
5.2.1.3 AasRoiOffsetX	64
5.2.1.4 AasRoiOffsetY	64
5.2.1.5 AasRoiWidth	65
5.2.1.6 AcquisitionAbort	65
5.2.1.7 AcquisitionArm	65
5.2.1.8 AcquisitionBurstFrameCount	65
5.2.1.9 AcquisitionFrameCount	65
5.2.1.10 AcquisitionFrameRate	65
5.2.1.11 AcquisitionFrameRateEnable	65
5.2.1.12 AcquisitionLineRate	65
5.2.1.13 AcquisitionMode	66
5.2.1.14 AcquisitionResultingFrameRate	66
5.2.1.15 AcquisitionStart	66
5.2.1.16 AcquisitionStatus	66
5.2.1.17 AcquisitionStatusSelector	66
5.2.1.18 AcquisitionStop	66
5.2.1.19 ActionDeviceKey	66
5.2.1.20 ActionGroupKey	66
5.2.1.21 ActionGroupMask	67

5.2.1.22 ActionQueueSize . . . . .	67
5.2.1.23 ActionSelector . . . . .	67
5.2.1.24 ActionUnconditionalMode . . . . .	67
5.2.1.25 AdaptiveCompressionEnable . . . . .	67
5.2.1.26 AdcBitDepth . . . . .	67
5.2.1.27 aPAUSEMACCtrlFramesReceived . . . . .	67
5.2.1.28 aPAUSEMACCtrlFramesTransmitted . . . . .	67
5.2.1.29 AutoAlgorithmSelector . . . . .	68
5.2.1.30 AutoExposureControlLoopDamping . . . . .	68
5.2.1.31 AutoExposureControlPriority . . . . .	68
5.2.1.32 AutoExposureEVCompensation . . . . .	68
5.2.1.33 AutoExposureExposureTimeLowerLimit . . . . .	68
5.2.1.34 AutoExposureExposureTimeUpperLimit . . . . .	68
5.2.1.35 AutoExposureGainLowerLimit . . . . .	68
5.2.1.36 AutoExposureGainUpperLimit . . . . .	68
5.2.1.37 AutoExposureGreyValueLowerLimit . . . . .	69
5.2.1.38 AutoExposureGreyValueUpperLimit . . . . .	69
5.2.1.39 AutoExposureLightingMode . . . . .	69
5.2.1.40 AutoExposureMeteringMode . . . . .	69
5.2.1.41 AutoExposureTargetGreyValue . . . . .	69
5.2.1.42 AutoExposureTargetGreyValueAuto . . . . .	69
5.2.1.43 BalanceRatio . . . . .	69
5.2.1.44 BalanceRatioSelector . . . . .	69
5.2.1.45 BalanceWhiteAuto . . . . .	70
5.2.1.46 BalanceWhiteAutoDamping . . . . .	70
5.2.1.47 BalanceWhiteAutoLowerLimit . . . . .	70
5.2.1.48 BalanceWhiteAutoProfile . . . . .	70
5.2.1.49 BalanceWhiteAutoUpperLimit . . . . .	70
5.2.1.50 BinningHorizontal . . . . .	70
5.2.1.51 BinningHorizontalMode . . . . .	70
5.2.1.52 BinningSelector . . . . .	70
5.2.1.53 BinningVertical . . . . .	71
5.2.1.54 BinningVerticalMode . . . . .	71
5.2.1.55 BlackLevel . . . . .	71
5.2.1.56 BlackLevelAuto . . . . .	71
5.2.1.57 BlackLevelAutoBalance . . . . .	71
5.2.1.58 BlackLevelClampingEnable . . . . .	71
5.2.1.59 BlackLevelRaw . . . . .	71
5.2.1.60 BlackLevelSelector . . . . .	71
5.2.1.61 ChunkBlackLevel . . . . .	72
5.2.1.62 ChunkBlackLevelSelector . . . . .	72
5.2.1.63 ChunkCounterSelector . . . . .	72

5.2.1.64 ChunkCounterValue . . . . .	72
5.2.1.65 ChunkCRC . . . . .	72
5.2.1.66 ChunkEnable . . . . .	72
5.2.1.67 ChunkEncoderSelector . . . . .	72
5.2.1.68 ChunkEncoderStatus . . . . .	72
5.2.1.69 ChunkEncoderValue . . . . .	73
5.2.1.70 ChunkExposureEndLineStatusAll . . . . .	73
5.2.1.71 ChunkExposureTime . . . . .	73
5.2.1.72 ChunkExposureTimeSelector . . . . .	73
5.2.1.73 ChunkFrameID . . . . .	73
5.2.1.74 ChunkGain . . . . .	73
5.2.1.75 ChunkGainSelector . . . . .	73
5.2.1.76 ChunkHeight . . . . .	73
5.2.1.77 ChunkImage . . . . .	74
5.2.1.78 ChunkImageComponent . . . . .	74
5.2.1.79 ChunkInferenceBoundingBoxResult . . . . .	74
5.2.1.80 ChunkInferenceConfidence . . . . .	74
5.2.1.81 ChunkInferenceFrameId . . . . .	74
5.2.1.82 ChunkInferenceResult . . . . .	74
5.2.1.83 ChunkLinePitch . . . . .	74
5.2.1.84 ChunkLineStatusAll . . . . .	74
5.2.1.85 ChunkModeActive . . . . .	75
5.2.1.86 ChunkOffsetX . . . . .	75
5.2.1.87 ChunkOffsetY . . . . .	75
5.2.1.88 ChunkPartSelector . . . . .	75
5.2.1.89 ChunkPixelDynamicRangeMax . . . . .	75
5.2.1.90 ChunkPixelDynamicRangeMin . . . . .	75
5.2.1.91 ChunkPixelFormat . . . . .	75
5.2.1.92 ChunkRegionID . . . . .	75
5.2.1.93 ChunkScan3dAxisMax . . . . .	76
5.2.1.94 ChunkScan3dAxisMin . . . . .	76
5.2.1.95 ChunkScan3dCoordinateOffset . . . . .	76
5.2.1.96 ChunkScan3dCoordinateReferenceSelector . . . . .	76
5.2.1.97 ChunkScan3dCoordinateReferenceValue . . . . .	76
5.2.1.98 ChunkScan3dCoordinateScale . . . . .	76
5.2.1.99 ChunkScan3dCoordinateSelector . . . . .	76
5.2.1.100 ChunkScan3dCoordinateSystem . . . . .	76
5.2.1.101 ChunkScan3dCoordinateSystemReference . . . . .	77
5.2.1.102 ChunkScan3dCoordinateTransformSelector . . . . .	77
5.2.1.103 ChunkScan3dDistanceUnit . . . . .	77
5.2.1.104 ChunkScan3dInvalidDataFlag . . . . .	77
5.2.1.105 ChunkScan3dInvalidDataValue . . . . .	77

---

5.2.1.106 ChunkScan3dOutputMode . . . . .	77
5.2.1.107 ChunkScan3dTransformValue . . . . .	77
5.2.1.108 ChunkScanLineSelector . . . . .	77
5.2.1.109 ChunkSelector . . . . .	78
5.2.1.110 ChunkSequencerSetActive . . . . .	78
5.2.1.111 ChunkSerialData . . . . .	78
5.2.1.112 ChunkSerialDataLength . . . . .	78
5.2.1.113 ChunkSerialReceiveOverflow . . . . .	78
5.2.1.114 ChunkSourceID . . . . .	78
5.2.1.115 ChunkStreamChannelID . . . . .	78
5.2.1.116 ChunkTimerSelector . . . . .	78
5.2.1.117 ChunkTimerValue . . . . .	79
5.2.1.118 ChunkTimestamp . . . . .	79
5.2.1.119 ChunkTimestampLatchValue . . . . .	79
5.2.1.120 ChunkTransferBlockID . . . . .	79
5.2.1.121 ChunkTransferQueueCurrentBlockCount . . . . .	79
5.2.1.122 ChunkTransferStreamID . . . . .	79
5.2.1.123 ChunkWidth . . . . .	79
5.2.1.124 CIConfiguration . . . . .	79
5.2.1.125 CITimeSlotsCount . . . . .	80
5.2.1.126 ColorTransformationEnable . . . . .	80
5.2.1.127 ColorTransformationSelector . . . . .	80
5.2.1.128 ColorTransformationValue . . . . .	80
5.2.1.129 ColorTransformationValueSelector . . . . .	80
5.2.1.130 CompressionRatio . . . . .	80
5.2.1.131 CounterDelay . . . . .	80
5.2.1.132 CounterDuration . . . . .	80
5.2.1.133 CounterEventActivation . . . . .	81
5.2.1.134 CounterEventSource . . . . .	81
5.2.1.135 CounterReset . . . . .	81
5.2.1.136 CounterResetActivation . . . . .	81
5.2.1.137 CounterResetSource . . . . .	81
5.2.1.138 CounterSelector . . . . .	81
5.2.1.139 CounterStatus . . . . .	81
5.2.1.140 CounterTriggerActivation . . . . .	81
5.2.1.141 CounterTriggerSource . . . . .	82
5.2.1.142 CounterValue . . . . .	82
5.2.1.143 CounterValueAtReset . . . . .	82
5.2.1.144 CxpConnectionSelector . . . . .	82
5.2.1.145 CxpConnectionTestErrorCount . . . . .	82
5.2.1.146 CxpConnectionTestMode . . . . .	82
5.2.1.147 CxpConnectionTestPacketCount . . . . .	82

5.2.1.148 CxpLinkConfiguration	82
5.2.1.149 CxpLinkConfigurationPreferred	83
5.2.1.150 CxpLinkConfigurationStatus	83
5.2.1.151 CxpPoCxpAuto	83
5.2.1.152 CxpPoCxpStatus	83
5.2.1.153 CxpPoCxpTripReset	83
5.2.1.154 CxpPoCxpTurnOff	83
5.2.1.155 DecimationHorizontal	83
5.2.1.156 DecimationHorizontalMode	83
5.2.1.157 DecimationSelector	84
5.2.1.158 DecimationVertical	84
5.2.1.159 DecimationVerticalMode	84
5.2.1.160 DefectCorrectionMode	84
5.2.1.161 DefectCorrectStaticEnable	84
5.2.1.162 DefectTableApply	84
5.2.1.163 DefectTableCoordinateX	84
5.2.1.164 DefectTableCoordinateY	84
5.2.1.165 DefectTableFactoryRestore	85
5.2.1.166 DefectTableIndex	85
5.2.1.167 DefectTablePixelCount	85
5.2.1.168 DefectTableSave	85
5.2.1.169 Deinterlacing	85
5.2.1.170 DeviceCharacterSet	85
5.2.1.171 DeviceClockFrequency	85
5.2.1.172 DeviceClockSelector	85
5.2.1.173 DeviceConnectionSelector	86
5.2.1.174 DeviceConnectionSpeed	86
5.2.1.175 DeviceConnectionStatus	86
5.2.1.176 DeviceEventChannelCount	86
5.2.1.177 DeviceFamilyName	86
5.2.1.178 DeviceFeaturePersistenceEnd	86
5.2.1.179 DeviceFeaturePersistenceStart	86
5.2.1.180 DeviceFirmwareVersion	86
5.2.1.181 DeviceGenCPVersionMajor	87
5.2.1.182 DeviceGenCPVersionMinor	87
5.2.1.183 DeviceID	87
5.2.1.184 DeviceIndicatorMode	87
5.2.1.185 DeviceLinkBandwidthReserve	87
5.2.1.186 DeviceLinkCommandTimeout	87
5.2.1.187 DeviceLinkConnectionCount	87
5.2.1.188 DeviceLinkCurrentThroughput	87
5.2.1.189 DeviceLinkHeartbeatMode	88



---

5.2.1.190 DeviceLinkHeartbeatTimeout . . . . .	88
5.2.1.191 DeviceLinkSelector . . . . .	88
5.2.1.192 DeviceLinkSpeed . . . . .	88
5.2.1.193 DeviceLinkThroughputLimit . . . . .	88
5.2.1.194 DeviceLinkThroughputLimitMode . . . . .	88
5.2.1.195 DeviceManifestEntrySelector . . . . .	88
5.2.1.196 DeviceManifestPrimaryURL . . . . .	88
5.2.1.197 DeviceManifestSchemaMajorVersion . . . . .	89
5.2.1.198 DeviceManifestSchemaMinorVersion . . . . .	89
5.2.1.199 DeviceManifestSecondaryURL . . . . .	89
5.2.1.200 DeviceManifestXMLMajorVersion . . . . .	89
5.2.1.201 DeviceManifestXMLMinorVersion . . . . .	89
5.2.1.202 DeviceManifestXMLSubMinorVersion . . . . .	89
5.2.1.203 DeviceManufacturerInfo . . . . .	89
5.2.1.204 DeviceMaxThroughput . . . . .	89
5.2.1.205 DeviceModelName . . . . .	90
5.2.1.206 DevicePowerSupplySelector . . . . .	90
5.2.1.207 DeviceRegistersCheck . . . . .	90
5.2.1.208 DeviceRegistersEndianness . . . . .	90
5.2.1.209 DeviceRegistersStreamingEnd . . . . .	90
5.2.1.210 DeviceRegistersStreamingStart . . . . .	90
5.2.1.211 DeviceRegistersValid . . . . .	90
5.2.1.212 DeviceReset . . . . .	90
5.2.1.213 DeviceScanType . . . . .	91
5.2.1.214 DeviceSerialNumber . . . . .	91
5.2.1.215 DeviceSerialPortBaudRate . . . . .	91
5.2.1.216 DeviceSerialPortSelector . . . . .	91
5.2.1.217 DeviceSFNCVersionMajor . . . . .	91
5.2.1.218 DeviceSFNCVersionMinor . . . . .	91
5.2.1.219 DeviceSFNCVersionSubMinor . . . . .	91
5.2.1.220 DeviceStreamChannelCount . . . . .	91
5.2.1.221 DeviceStreamChannelEndianness . . . . .	92
5.2.1.222 DeviceStreamChannelLink . . . . .	92
5.2.1.223 DeviceStreamChannelPacketSize . . . . .	92
5.2.1.224 DeviceStreamChannelSelector . . . . .	92
5.2.1.225 DeviceStreamChannelType . . . . .	92
5.2.1.226 DeviceTapGeometry . . . . .	92
5.2.1.227 DeviceTemperature . . . . .	92
5.2.1.228 DeviceTemperatureSelector . . . . .	92
5.2.1.229 DeviceTLType . . . . .	93
5.2.1.230 DeviceTLVersionMajor . . . . .	93
5.2.1.231 DeviceTLVersionMinor . . . . .	93

5.2.1.232 DeviceTLVersionSubMinor . . . . .	93
5.2.1.233 DeviceType . . . . .	93
5.2.1.234 DeviceUptime . . . . .	93
5.2.1.235 DeviceUserID . . . . .	93
5.2.1.236 DeviceVendorName . . . . .	93
5.2.1.237 DeviceVersion . . . . .	94
5.2.1.238 EncoderDivider . . . . .	94
5.2.1.239 EncoderMode . . . . .	94
5.2.1.240 EncoderOutputMode . . . . .	94
5.2.1.241 EncoderReset . . . . .	94
5.2.1.242 EncoderResetActivation . . . . .	94
5.2.1.243 EncoderResetSource . . . . .	94
5.2.1.244 EncoderSelector . . . . .	94
5.2.1.245 EncoderSourceA . . . . .	95
5.2.1.246 EncoderSourceB . . . . .	95
5.2.1.247 EncoderStatus . . . . .	95
5.2.1.248 EncoderTimeout . . . . .	95
5.2.1.249 EncoderValue . . . . .	95
5.2.1.250 EncoderValueAtReset . . . . .	95
5.2.1.251 EnumerationCount . . . . .	95
5.2.1.252 EventAcquisitionEnd . . . . .	95
5.2.1.253 EventAcquisitionEndFrameID . . . . .	96
5.2.1.254 EventAcquisitionEndTimestamp . . . . .	96
5.2.1.255 EventAcquisitionError . . . . .	96
5.2.1.256 EventAcquisitionErrorFrameID . . . . .	96
5.2.1.257 EventAcquisitionErrorTimestamp . . . . .	96
5.2.1.258 EventAcquisitionStart . . . . .	96
5.2.1.259 EventAcquisitionStartFrameID . . . . .	96
5.2.1.260 EventAcquisitionStartTimestamp . . . . .	96
5.2.1.261 EventAcquisitionTransferEnd . . . . .	97
5.2.1.262 EventAcquisitionTransferEndFrameID . . . . .	97
5.2.1.263 EventAcquisitionTransferEndTimestamp . . . . .	97
5.2.1.264 EventAcquisitionTransferStart . . . . .	97
5.2.1.265 EventAcquisitionTransferStartFrameID . . . . .	97
5.2.1.266 EventAcquisitionTransferStartTimestamp . . . . .	97
5.2.1.267 EventAcquisitionTrigger . . . . .	97
5.2.1.268 EventAcquisitionTriggerFrameID . . . . .	97
5.2.1.269 EventAcquisitionTriggerTimestamp . . . . .	98
5.2.1.270 EventActionLate . . . . .	98
5.2.1.271 EventActionLateFrameID . . . . .	98
5.2.1.272 EventActionLateTimestamp . . . . .	98
5.2.1.273 EventCounter0End . . . . .	98

---

5.2.1.274 EventCounter0EndFrameID . . . . .	98
5.2.1.275 EventCounter0EndTimestamp . . . . .	98
5.2.1.276 EventCounter0Start . . . . .	98
5.2.1.277 EventCounter0StartFrameID . . . . .	99
5.2.1.278 EventCounter0StartTimestamp . . . . .	99
5.2.1.279 EventCounter1End . . . . .	99
5.2.1.280 EventCounter1EndFrameID . . . . .	99
5.2.1.281 EventCounter1EndTimestamp . . . . .	99
5.2.1.282 EventCounter1Start . . . . .	99
5.2.1.283 EventCounter1StartFrameID . . . . .	99
5.2.1.284 EventCounter1StartTimestamp . . . . .	99
5.2.1.285 EventEncoder0Restarted . . . . .	100
5.2.1.286 EventEncoder0RestartedFrameID . . . . .	100
5.2.1.287 EventEncoder0RestartedTimestamp . . . . .	100
5.2.1.288 EventEncoder0Stopped . . . . .	100
5.2.1.289 EventEncoder0StoppedFrameID . . . . .	100
5.2.1.290 EventEncoder0StoppedTimestamp . . . . .	100
5.2.1.291 EventEncoder1Restarted . . . . .	100
5.2.1.292 EventEncoder1RestartedFrameID . . . . .	100
5.2.1.293 EventEncoder1RestartedTimestamp . . . . .	101
5.2.1.294 EventEncoder1Stopped . . . . .	101
5.2.1.295 EventEncoder1StoppedFrameID . . . . .	101
5.2.1.296 EventEncoder1StoppedTimestamp . . . . .	101
5.2.1.297 EventError . . . . .	101
5.2.1.298 EventErrorCode . . . . .	101
5.2.1.299 EventErrorFrameID . . . . .	101
5.2.1.300 EventErrorTimestamp . . . . .	101
5.2.1.301 EventExposureEnd . . . . .	102
5.2.1.302 EventExposureEndFrameID . . . . .	102
5.2.1.303 EventExposureEndTimestamp . . . . .	102
5.2.1.304 EventExposureStart . . . . .	102
5.2.1.305 EventExposureStartFrameID . . . . .	102
5.2.1.306 EventExposureStartTimestamp . . . . .	102
5.2.1.307 EventFrameBurstEnd . . . . .	102
5.2.1.308 EventFrameBurstEndFrameID . . . . .	102
5.2.1.309 EventFrameBurstEndTimestamp . . . . .	103
5.2.1.310 EventFrameBurstStart . . . . .	103
5.2.1.311 EventFrameBurstStartFrameID . . . . .	103
5.2.1.312 EventFrameBurstStartTimestamp . . . . .	103
5.2.1.313 EventFrameEnd . . . . .	103
5.2.1.314 EventFrameEndFrameID . . . . .	103
5.2.1.315 EventFrameEndTimestamp . . . . .	103

5.2.1.316 EventFrameStart . . . . .	103
5.2.1.317 EventFrameStartFrameID . . . . .	104
5.2.1.318 EventFrameStartTimestamp . . . . .	104
5.2.1.319 EventFrameTransferEnd . . . . .	104
5.2.1.320 EventFrameTransferEndFrameID . . . . .	104
5.2.1.321 EventFrameTransferEndTimestamp . . . . .	104
5.2.1.322 EventFrameTransferStart . . . . .	104
5.2.1.323 EventFrameTransferStartFrameID . . . . .	104
5.2.1.324 EventFrameTransferStartTimestamp . . . . .	104
5.2.1.325 EventFrameTrigger . . . . .	105
5.2.1.326 EventFrameTriggerFrameID . . . . .	105
5.2.1.327 EventFrameTriggerTimestamp . . . . .	105
5.2.1.328 EventLine0AnyEdge . . . . .	105
5.2.1.329 EventLine0AnyEdgeFrameID . . . . .	105
5.2.1.330 EventLine0AnyEdgeTimestamp . . . . .	105
5.2.1.331 EventLine0FallingEdge . . . . .	105
5.2.1.332 EventLine0FallingEdgeFrameID . . . . .	105
5.2.1.333 EventLine0FallingEdgeTimestamp . . . . .	106
5.2.1.334 EventLine0RisingEdge . . . . .	106
5.2.1.335 EventLine0RisingEdgeFrameID . . . . .	106
5.2.1.336 EventLine0RisingEdgeTimestamp . . . . .	106
5.2.1.337 EventLine1AnyEdge . . . . .	106
5.2.1.338 EventLine1AnyEdgeFrameID . . . . .	106
5.2.1.339 EventLine1AnyEdgeTimestamp . . . . .	106
5.2.1.340 EventLine1FallingEdge . . . . .	106
5.2.1.341 EventLine1FallingEdgeFrameID . . . . .	107
5.2.1.342 EventLine1FallingEdgeTimestamp . . . . .	107
5.2.1.343 EventLine1RisingEdge . . . . .	107
5.2.1.344 EventLine1RisingEdgeFrameID . . . . .	107
5.2.1.345 EventLine1RisingEdgeTimestamp . . . . .	107
5.2.1.346 EventLinkSpeedChange . . . . .	107
5.2.1.347 EventLinkSpeedChangeFrameID . . . . .	107
5.2.1.348 EventLinkSpeedChangeTimestamp . . . . .	107
5.2.1.349 EventLinkTrigger0 . . . . .	108
5.2.1.350 EventLinkTrigger0FrameID . . . . .	108
5.2.1.351 EventLinkTrigger0Timestamp . . . . .	108
5.2.1.352 EventLinkTrigger1 . . . . .	108
5.2.1.353 EventLinkTrigger1FrameID . . . . .	108
5.2.1.354 EventLinkTrigger1Timestamp . . . . .	108
5.2.1.355 EventNotification . . . . .	108
5.2.1.356 EventSelector . . . . .	108
5.2.1.357 EventSequencerSetChange . . . . .	109

---

5.2.1.358 EventSequencerSetChangeFrameID . . . . .	109
5.2.1.359 EventSequencerSetChangeTimestamp . . . . .	109
5.2.1.360 EventSerialData . . . . .	109
5.2.1.361 EventSerialDataLength . . . . .	109
5.2.1.362 EventSerialPortReceive . . . . .	109
5.2.1.363 EventSerialPortReceiveTimestamp . . . . .	109
5.2.1.364 EventSerialReceiveOverflow . . . . .	109
5.2.1.365 EventStream0TransferBlockEnd . . . . .	110
5.2.1.366 EventStream0TransferBlockEndFrameID . . . . .	110
5.2.1.367 EventStream0TransferBlockEndTimestamp . . . . .	110
5.2.1.368 EventStream0TransferBlockStart . . . . .	110
5.2.1.369 EventStream0TransferBlockStartFrameID . . . . .	110
5.2.1.370 EventStream0TransferBlockStartTimestamp . . . . .	110
5.2.1.371 EventStream0TransferBlockTrigger . . . . .	110
5.2.1.372 EventStream0TransferBlockTriggerFrameID . . . . .	110
5.2.1.373 EventStream0TransferBlockTriggerTimestamp . . . . .	111
5.2.1.374 EventStream0TransferBurstEnd . . . . .	111
5.2.1.375 EventStream0TransferBurstEndFrameID . . . . .	111
5.2.1.376 EventStream0TransferBurstEndTimestamp . . . . .	111
5.2.1.377 EventStream0TransferBurstStart . . . . .	111
5.2.1.378 EventStream0TransferBurstStartFrameID . . . . .	111
5.2.1.379 EventStream0TransferBurstStartTimestamp . . . . .	111
5.2.1.380 EventStream0TransferEnd . . . . .	111
5.2.1.381 EventStream0TransferEndFrameID . . . . .	112
5.2.1.382 EventStream0TransferEndTimestamp . . . . .	112
5.2.1.383 EventStream0TransferOverflow . . . . .	112
5.2.1.384 EventStream0TransferOverflowFrameID . . . . .	112
5.2.1.385 EventStream0TransferOverflowTimestamp . . . . .	112
5.2.1.386 EventStream0TransferPause . . . . .	112
5.2.1.387 EventStream0TransferPauseFrameID . . . . .	112
5.2.1.388 EventStream0TransferPauseTimestamp . . . . .	112
5.2.1.389 EventStream0TransferResume . . . . .	113
5.2.1.390 EventStream0TransferResumeFrameID . . . . .	113
5.2.1.391 EventStream0TransferResumeTimestamp . . . . .	113
5.2.1.392 EventStream0TransferStart . . . . .	113
5.2.1.393 EventStream0TransferStartFrameID . . . . .	113
5.2.1.394 EventStream0TransferStartTimestamp . . . . .	113
5.2.1.395 EventTest . . . . .	113
5.2.1.396 EventTestTimestamp . . . . .	113
5.2.1.397 EventTimer0End . . . . .	114
5.2.1.398 EventTimer0EndFrameID . . . . .	114
5.2.1.399 EventTimer0EndTimestamp . . . . .	114

5.2.1.400 EventTimer0Start	114
5.2.1.401 EventTimer0StartFrameID	114
5.2.1.402 EventTimer0StartTimestamp	114
5.2.1.403 EventTimer1End	114
5.2.1.404 EventTimer1EndFrameID	114
5.2.1.405 EventTimer1EndTimestamp	115
5.2.1.406 EventTimer1Start	115
5.2.1.407 EventTimer1StartFrameID	115
5.2.1.408 EventTimer1StartTimestamp	115
5.2.1.409 ExposureActiveMode	115
5.2.1.410 ExposureAuto	115
5.2.1.411 ExposureMode	115
5.2.1.412 ExposureTime	115
5.2.1.413 ExposureTimeMode	116
5.2.1.414 ExposureTimeSelector	116
5.2.1.415 FactoryReset	116
5.2.1.416 FileAccessBuffer	116
5.2.1.417 FileAccessLength	116
5.2.1.418 FileAccessOffset	116
5.2.1.419 FileOpenMode	116
5.2.1.420 FileOperationExecute	116
5.2.1.421 FileOperationResult	117
5.2.1.422 FileOperationSelector	117
5.2.1.423 FileOperationStatus	117
5.2.1.424 FileSelector	117
5.2.1.425 FileSize	117
5.2.1.426 Gain	117
5.2.1.427 GainAuto	117
5.2.1.428 GainAutoBalance	117
5.2.1.429 GainSelector	118
5.2.1.430 Gamma	118
5.2.1.431 GammaEnable	118
5.2.1.432 GevActiveLinkCount	118
5.2.1.433 GevCCP	118
5.2.1.434 GevCurrentDefaultGateway	118
5.2.1.435 GevCurrentIPAddress	118
5.2.1.436 GevCurrentIPConfigurationDHCP	118
5.2.1.437 GevCurrentIPConfigurationLLA	119
5.2.1.438 GevCurrentIPConfigurationPersistentIP	119
5.2.1.439 GevCurrentPhysicalLinkConfiguration	119
5.2.1.440 GevCurrentSubnetMask	119
5.2.1.441 GevDiscoveryAckDelay	119

5.2.1.442	GevFirstURL	119
5.2.1.443	GevGVCPExtendedStatusCodes	119
5.2.1.444	GevGVCPExtendedStatusCodesSelector	119
5.2.1.445	GevGVCPHeartbeatDisable	120
5.2.1.446	GevGVCPPendingAck	120
5.2.1.447	GevGVCPPendingTimeout	120
5.2.1.448	GevGVSPExtendedIDMode	120
5.2.1.449	GevHeartbeatTimeout	120
5.2.1.450	GevIEEE1588	120
5.2.1.451	GevIEEE1588ClockAccuracy	120
5.2.1.452	GevIEEE1588Mode	120
5.2.1.453	GevIEEE1588Status	121
5.2.1.454	GevInterfaceSelector	121
5.2.1.455	GevIPConfigurationStatus	121
5.2.1.456	GevMACAddress	121
5.2.1.457	GevMCDA	121
5.2.1.458	GevMCPHostPort	121
5.2.1.459	GevMCRC	121
5.2.1.460	GevMCSP	121
5.2.1.461	GevMCTT	122
5.2.1.462	GevNumberOfInterfaces	122
5.2.1.463	GevPAUSEFrameReception	122
5.2.1.464	GevPAUSEFrameTransmission	122
5.2.1.465	GevPersistentDefaultGateway	122
5.2.1.466	GevPersistentIPAddress	122
5.2.1.467	GevPersistentSubnetMask	122
5.2.1.468	GevPhysicalLinkConfiguration	122
5.2.1.469	GevPrimaryApplicationIPAddress	123
5.2.1.470	GevPrimaryApplicationSocket	123
5.2.1.471	GevPrimaryApplicationSwitchoverKey	123
5.2.1.472	GevSCCFGAllInTransmission	123
5.2.1.473	GevSCCFGExtendedChunkData	123
5.2.1.474	GevSCCFGPacketResendDestination	123
5.2.1.475	GevSCCFGUnconditionalStreaming	123
5.2.1.476	GevSCDA	123
5.2.1.477	GevSCPD	124
5.2.1.478	GevSCPDirection	124
5.2.1.479	GevSCPHostPort	124
5.2.1.480	GevSCPInterfaceIndex	124
5.2.1.481	GevSCPSBigEndian	124
5.2.1.482	GevSCPSDoNotFragment	124
5.2.1.483	GevSCPSFireTestPacket	124

5.2.1.484	GevSCPSPacketSize	124
5.2.1.485	GevSCSP	125
5.2.1.486	GevSCZoneConfigurationLock	125
5.2.1.487	GevSCZoneCount	125
5.2.1.488	GevSCZoneDirectionAll	125
5.2.1.489	GevSecondURL	125
5.2.1.490	GevStreamChannelSelector	125
5.2.1.491	GevSupportedOption	125
5.2.1.492	GevSupportedOptionSelector	125
5.2.1.493	GevTimestampTickFrequency	126
5.2.1.494	GuiXmlManifestAddress	126
5.2.1.495	Height	126
5.2.1.496	HeightMax	126
5.2.1.497	ImageComponentEnable	126
5.2.1.498	ImageComponentSelector	126
5.2.1.499	ImageCompressionBitrate	126
5.2.1.500	ImageCompressionJPEGFormatOption	126
5.2.1.501	ImageCompressionMode	127
5.2.1.502	ImageCompressionQuality	127
5.2.1.503	ImageCompressionRateOption	127
5.2.1.504	IspEnable	127
5.2.1.505	LineFilterWidth	127
5.2.1.506	LineFormat	127
5.2.1.507	LineInputFilterSelector	127
5.2.1.508	LineInverter	127
5.2.1.509	LineMode	128
5.2.1.510	LinePitch	128
5.2.1.511	LineSelector	128
5.2.1.512	LineSource	128
5.2.1.513	LineStatus	128
5.2.1.514	LineStatusAll	128
5.2.1.515	LinkErrorCount	128
5.2.1.516	LinkUptime	128
5.2.1.517	LogicBlockLUTInputActivation	129
5.2.1.518	LogicBlockLUTInputSelector	129
5.2.1.519	LogicBlockLUTInputSource	129
5.2.1.520	LogicBlockLUTOutputValue	129
5.2.1.521	LogicBlockLUTOutputValueAll	129
5.2.1.522	LogicBlockLUTRowIndex	129
5.2.1.523	LogicBlockLUTSelector	129
5.2.1.524	LogicBlockSelector	129
5.2.1.525	LUTEnable	130



---

5.2.1.526 LUTIndex . . . . .	130
5.2.1.527 LUTSelector . . . . .	130
5.2.1.528 LUTValue . . . . .	130
5.2.1.529 LUTValueAll . . . . .	130
5.2.1.530 MaxDeviceResetTime . . . . .	130
5.2.1.531 OffsetX . . . . .	130
5.2.1.532 OffsetY . . . . .	130
5.2.1.533 PacketResendRequestCount . . . . .	131
5.2.1.534 PayloadSize . . . . .	131
5.2.1.535 PixelColorFilter . . . . .	131
5.2.1.536 PixelDynamicRangeMax . . . . .	131
5.2.1.537 PixelDynamicRangeMin . . . . .	131
5.2.1.538 PixelFormat . . . . .	131
5.2.1.539 PixelFormatInfoID . . . . .	131
5.2.1.540 PixelFormatInfoSelector . . . . .	131
5.2.1.541 PixelSize . . . . .	132
5.2.1.542 PowerSupplyCurrent . . . . .	132
5.2.1.543 PowerSupplyVoltage . . . . .	132
5.2.1.544 RegionDestination . . . . .	132
5.2.1.545 RegionMode . . . . .	132
5.2.1.546 RegionSelector . . . . .	132
5.2.1.547 ReverseX . . . . .	132
5.2.1.548 ReverseY . . . . .	132
5.2.1.549 RgbTransformLightSource . . . . .	133
5.2.1.550 Saturation . . . . .	133
5.2.1.551 SaturationEnable . . . . .	133
5.2.1.552 Scan3dAxisMax . . . . .	133
5.2.1.553 Scan3dAxisMin . . . . .	133
5.2.1.554 Scan3dCoordinateOffset . . . . .	133
5.2.1.555 Scan3dCoordinateReferenceSelector . . . . .	133
5.2.1.556 Scan3dCoordinateReferenceValue . . . . .	133
5.2.1.557 Scan3dCoordinateScale . . . . .	134
5.2.1.558 Scan3dCoordinateSelector . . . . .	134
5.2.1.559 Scan3dCoordinateSystem . . . . .	134
5.2.1.560 Scan3dCoordinateSystemReference . . . . .	134
5.2.1.561 Scan3dCoordinateTransformSelector . . . . .	134
5.2.1.562 Scan3dDistanceUnit . . . . .	134
5.2.1.563 Scan3dInvalidDataFlag . . . . .	134
5.2.1.564 Scan3dInvalidDataValue . . . . .	134
5.2.1.565 Scan3dOutputMode . . . . .	135
5.2.1.566 Scan3dTransformValue . . . . .	135
5.2.1.567 SensorDescription . . . . .	135

5.2.1.568 SensorDigitizationTaps . . . . .	135
5.2.1.569 SensorHeight . . . . .	135
5.2.1.570 SensorShutterMode . . . . .	135
5.2.1.571 SensorTaps . . . . .	135
5.2.1.572 SensorWidth . . . . .	135
5.2.1.573 SequencerConfigurationMode . . . . .	136
5.2.1.574 SequencerConfigurationValid . . . . .	136
5.2.1.575 SequencerFeatureEnable . . . . .	136
5.2.1.576 SequencerMode . . . . .	136
5.2.1.577 SequencerPathSelector . . . . .	136
5.2.1.578 SequencerSetActive . . . . .	136
5.2.1.579 SequencerSetLoad . . . . .	136
5.2.1.580 SequencerSetNext . . . . .	136
5.2.1.581 SequencerSetSave . . . . .	137
5.2.1.582 SequencerSetSelector . . . . .	137
5.2.1.583 SequencerSetStart . . . . .	137
5.2.1.584 SequencerSetValid . . . . .	137
5.2.1.585 SequencerTriggerActivation . . . . .	137
5.2.1.586 SequencerTriggerSource . . . . .	137
5.2.1.587 SerialPortBaudRate . . . . .	137
5.2.1.588 SerialPortDataBits . . . . .	137
5.2.1.589 SerialPortParity . . . . .	138
5.2.1.590 SerialPortSelector . . . . .	138
5.2.1.591 SerialPortSource . . . . .	138
5.2.1.592 SerialPortStopBits . . . . .	138
5.2.1.593 SerialReceiveFramingErrorCount . . . . .	138
5.2.1.594 SerialReceiveParityErrorCount . . . . .	138
5.2.1.595 SerialReceiveQueueClear . . . . .	138
5.2.1.596 SerialReceiveQueueCurrentCharacterCount . . . . .	138
5.2.1.597 SerialReceiveQueueMaxCharacterCount . . . . .	139
5.2.1.598 SerialTransmitQueueCurrentCharacterCount . . . . .	139
5.2.1.599 SerialTransmitQueueMaxCharacterCount . . . . .	139
5.2.1.600 Sharpening . . . . .	139
5.2.1.601 SharpeningAuto . . . . .	139
5.2.1.602 SharpeningEnable . . . . .	139
5.2.1.603 SharpeningThreshold . . . . .	139
5.2.1.604 SoftwareSignalPulse . . . . .	139
5.2.1.605 SoftwareSignalSelector . . . . .	140
5.2.1.606 SourceCount . . . . .	140
5.2.1.607 SourceSelector . . . . .	140
5.2.1.608 Test0001 . . . . .	140
5.2.1.609 TestEventGenerate . . . . .	140

---

5.2.1.610 TestPattern . . . . .	140
5.2.1.611 TestPatternGeneratorSelector . . . . .	140
5.2.1.612 TestPendingAck . . . . .	140
5.2.1.613 TimerDelay . . . . .	141
5.2.1.614 TimerDuration . . . . .	141
5.2.1.615 TimerReset . . . . .	141
5.2.1.616 TimerSelector . . . . .	141
5.2.1.617 TimerStatus . . . . .	141
5.2.1.618 TimerTriggerActivation . . . . .	141
5.2.1.619 TimerTriggerSource . . . . .	141
5.2.1.620 TimerValue . . . . .	141
5.2.1.621 Timestamp . . . . .	142
5.2.1.622 TimestampLatch . . . . .	142
5.2.1.623 TimestampLatchValue . . . . .	142
5.2.1.624 TimestampReset . . . . .	142
5.2.1.625 TLParamsLocked . . . . .	142
5.2.1.626 TransferAbort . . . . .	142
5.2.1.627 TransferBlockCount . . . . .	142
5.2.1.628 TransferBurstCount . . . . .	142
5.2.1.629 TransferComponentSelector . . . . .	143
5.2.1.630 TransferControlMode . . . . .	143
5.2.1.631 TransferOperationMode . . . . .	143
5.2.1.632 TransferPause . . . . .	143
5.2.1.633 TransferQueueCurrentBlockCount . . . . .	143
5.2.1.634 TransferQueueMaxBlockCount . . . . .	143
5.2.1.635 TransferQueueMode . . . . .	143
5.2.1.636 TransferQueueOverflowCount . . . . .	143
5.2.1.637 TransferResume . . . . .	144
5.2.1.638 TransferSelector . . . . .	144
5.2.1.639 TransferStart . . . . .	144
5.2.1.640 TransferStatus . . . . .	144
5.2.1.641 TransferStatusSelector . . . . .	144
5.2.1.642 TransferStop . . . . .	144
5.2.1.643 TransferStreamChannel . . . . .	144
5.2.1.644 TransferTriggerActivation . . . . .	144
5.2.1.645 TransferTriggerMode . . . . .	145
5.2.1.646 TransferTriggerSelector . . . . .	145
5.2.1.647 TransferTriggerSource . . . . .	145
5.2.1.648 TriggerActivation . . . . .	145
5.2.1.649 TriggerDelay . . . . .	145
5.2.1.650 TriggerDivider . . . . .	145
5.2.1.651 TriggerEventTest . . . . .	145

5.2.1.652 TriggerMode	145
5.2.1.653 TriggerMultiplier	146
5.2.1.654 TriggerOverlap	146
5.2.1.655 TriggerSelector	146
5.2.1.656 TriggerSoftware	146
5.2.1.657 TriggerSource	146
5.2.1.658 UserOutputSelector	146
5.2.1.659 UserOutputValue	146
5.2.1.660 UserOutputValueAll	146
5.2.1.661 UserOutputValueAllMask	147
5.2.1.662 UserSetDefault	147
5.2.1.663 UserSetFeatureEnable	147
5.2.1.664 UserSetLoad	147
5.2.1.665 UserSetSave	147
5.2.1.666 UserSetSelector	147
5.2.1.667 V3_3Enable	147
5.2.1.668 WhiteClip	147
5.2.1.669 WhiteClipSelector	148
5.2.1.670 Width	148
5.2.1.671 WidthMax	148
5.3 _quickSpinTLDevice Struct Reference	148
5.3.1 Field Documentation	149
5.3.1.1 DeviceAccessStatus	149
5.3.1.2 DeviceCurrentSpeed	149
5.3.1.3 DeviceDisplayName	149
5.3.1.4 DeviceDriverVersion	149
5.3.1.5 DeviceEndianessMechanism	150
5.3.1.6 DeviceID	150
5.3.1.7 DeviceInstanceld	150
5.3.1.8 DeviceIsUpdater	150
5.3.1.9 DeviceLinkSpeed	150
5.3.1.10 DeviceLocation	150
5.3.1.11 DeviceModelName	150
5.3.1.12 DeviceMulticastMonitorMode	150
5.3.1.13 DeviceSerialNumber	151
5.3.1.14 DeviceType	151
5.3.1.15 DeviceU3VProtocol	151
5.3.1.16 DeviceUserID	151
5.3.1.17 DeviceVendorName	151
5.3.1.18 DeviceVersion	151
5.3.1.19 GenICamXMLLocation	151
5.3.1.20 GenICamXMLPath	151

5.3.1.21	GevCCP	152
5.3.1.22	GevDeviceAutoForceIP	152
5.3.1.23	GevDeviceDiscoverMaximumPacketSize	152
5.3.1.24	GevDeviceForceGateway	152
5.3.1.25	GevDeviceForceIP	152
5.3.1.26	GevDeviceForceIPAddress	152
5.3.1.27	GevDeviceForceSubnetMask	152
5.3.1.28	GevDeviceGateway	152
5.3.1.29	GevDeviceIPAddress	153
5.3.1.30	GevDevicesWrongSubnet	153
5.3.1.31	GevDeviceMACAddress	153
5.3.1.32	GevDeviceMaximumPacketSize	153
5.3.1.33	GevDeviceMaximumRetryCount	153
5.3.1.34	GevDeviceModelsBigEndian	153
5.3.1.35	GevDevicePort	153
5.3.1.36	GevDeviceReadAndWriteTimeout	153
5.3.1.37	GevDeviceSubnetMask	154
5.3.1.38	GevVersionMajor	154
5.3.1.39	GevVersionMinor	154
5.3.1.40	GUIXMLLocation	154
5.3.1.41	GUIXMLPath	154
5.4	_quickSpinTLInterface Struct Reference	154
5.4.1	Field Documentation	155
5.4.1.1	ActionCommand	155
5.4.1.2	DeviceAccessStatus	155
5.4.1.3	DeviceCount	156
5.4.1.4	DeviceID	156
5.4.1.5	DeviceModelName	156
5.4.1.6	DeviceSelector	156
5.4.1.7	DeviceSerialNumber	156
5.4.1.8	DeviceUnlock	156
5.4.1.9	DeviceUpdateList	156
5.4.1.10	DeviceVendorName	156
5.4.1.11	FilterDriverStatus	157
5.4.1.12	GevActionDeviceKey	157
5.4.1.13	GevActionGroupKey	157
5.4.1.14	GevActionGroupMask	157
5.4.1.15	GevActionTime	157
5.4.1.16	GevDeviceAutoForceIP	157
5.4.1.17	GevDeviceForceGateway	157
5.4.1.18	GevDeviceForceIP	157
5.4.1.19	GevDeviceForceIPAddress	158

5.4.1.20	GevDeviceForceSubnetMask	158
5.4.1.21	GevDeviceGateway	158
5.4.1.22	GevDeviceIPAddress	158
5.4.1.23	GevDeviceMACAddress	158
5.4.1.24	GevDeviceSubnetMask	158
5.4.1.25	GevInterfaceGateway	158
5.4.1.26	GevInterfaceGatewaySelector	158
5.4.1.27	GevInterfaceMACAddress	159
5.4.1.28	GevInterfaceMTU	159
5.4.1.29	GevInterfaceReceiveLinkSpeed	159
5.4.1.30	GevInterfaceSubnetIPAddress	159
5.4.1.31	GevInterfaceSubnetMask	159
5.4.1.32	GevInterfaceSubnetSelector	159
5.4.1.33	GevInterfaceTransmitLinkSpeed	159
5.4.1.34	HostAdapterDriverVersion	159
5.4.1.35	HostAdapterName	160
5.4.1.36	HostAdapterVendor	160
5.4.1.37	IncompatibleDeviceCount	160
5.4.1.38	IncompatibleDeviceID	160
5.4.1.39	IncompatibleDeviceModelName	160
5.4.1.40	IncompatibleDeviceSelector	160
5.4.1.41	IncompatibleDeviceVendorName	160
5.4.1.42	IncompatibleGevDeviceIPAddress	160
5.4.1.43	IncompatibleGevDeviceMACAddress	161
5.4.1.44	IncompatibleGevDeviceSubnetMask	161
5.4.1.45	InterfaceDisplayName	161
5.4.1.46	InterfaceID	161
5.4.1.47	InterfaceType	161
5.4.1.48	POEStatus	161
5.5	_quickSpinTLStream Struct Reference	162
5.5.1	Field Documentation	162
5.5.1.1	GevFailedPacketCount	162
5.5.1.2	GevMaximumNumberResendRequests	162
5.5.1.3	GevPacketResendMode	163
5.5.1.4	GevPacketResendTimeout	163
5.5.1.5	GevResendPacketCount	163
5.5.1.6	GevResendRequestCount	163
5.5.1.7	GevTotalPacketCount	163
5.5.1.8	StreamAnnounceBufferMinimum	163
5.5.1.9	StreamAnnouncedBufferCount	163
5.5.1.10	StreamBlockTransferSize	163
5.5.1.11	StreamBufferAlignment	164

5.5.1.12 StreamBufferCountManual	164
5.5.1.13 StreamBufferCountMax	164
5.5.1.14 StreamBufferCountMode	164
5.5.1.15 StreamBufferCountResult	164
5.5.1.16 StreamBufferHandlingMode	164
5.5.1.17 StreamChunkCountMaximum	164
5.5.1.18 StreamCRCCheckEnable	164
5.5.1.19 StreamDeliveredFrameCount	165
5.5.1.20 StreamFailedBufferCount	165
5.5.1.21 StreamID	165
5.5.1.22 StreamInputBufferCount	165
5.5.1.23 StreamIsGrabbing	165
5.5.1.24 StreamLostFrameCount	165
5.5.1.25 StreamOutputBufferCount	165
5.5.1.26 StreamStartedFrameCount	165
5.5.1.27 StreamType	166
5.6 _quickSpinTLSystem Struct Reference	166
5.6.1 Field Documentation	166
5.6.1.1 EnumerateGEVInterfaces	166
5.6.1.2 GenTLFSFNCVersionMajor	167
5.6.1.3 GenTLFSFNCVersionMinor	167
5.6.1.4 GenTLFSFNCVersionSubMinor	167
5.6.1.5 GenTLVersionMajor	167
5.6.1.6 GenTLVersionMinor	167
5.6.1.7 GevInterfaceDefaultGateway	167
5.6.1.8 GevInterfaceDefaultIPAddress	167
5.6.1.9 GevInterfaceDefaultSubnetMask	167
5.6.1.10 GevInterfaceMACAddress	168
5.6.1.11 GevVersionMajor	168
5.6.1.12 GevVersionMinor	168
5.6.1.13 InterfaceDisplayName	168
5.6.1.14 InterfaceID	168
5.6.1.15 InterfaceSelector	168
5.6.1.16 InterfaceUpdateList	168
5.6.1.17 TLDisplayName	168
5.6.1.18 TLFileName	169
5.6.1.19 TLID	169
5.6.1.20 TLModelName	169
5.6.1.21 TLPath	169
5.6.1.22 TLType	169
5.6.1.23 TLVendorName	169
5.6.1.24 TLVersion	169

5.7 _spinAVIOption Struct Reference . . . . .	170
5.7.1 Detailed Description . . . . .	170
5.7.2 Field Documentation . . . . .	170
5.7.2.1 frameRate . . . . .	170
5.7.2.2 reserved . . . . .	170
5.8 _spinBMPOption Struct Reference . . . . .	170
5.8.1 Detailed Description . . . . .	171
5.8.2 Field Documentation . . . . .	171
5.8.2.1 indexedColor_8bit . . . . .	171
5.8.2.2 reserved . . . . .	171
5.9 _spinChunkData Struct Reference . . . . .	171
5.9.1 Detailed Description . . . . .	172
5.9.2 Field Documentation . . . . .	172
5.9.2.1 m_blackLevel . . . . .	172
5.9.2.2 m_counterValue . . . . .	173
5.9.2.3 m_cRC . . . . .	173
5.9.2.4 m_encoderValue . . . . .	173
5.9.2.5 m_exposureEndLineStatusAll . . . . .	173
5.9.2.6 m_exposureTime . . . . .	173
5.9.2.7 m_frameID . . . . .	173
5.9.2.8 m_gain . . . . .	173
5.9.2.9 m_height . . . . .	173
5.9.2.10 m_image . . . . .	174
5.9.2.11 m_inferenceConfidence . . . . .	174
5.9.2.12 m_inferenceFrameID . . . . .	174
5.9.2.13 m_inferenceResult . . . . .	174
5.9.2.14 m_linePitch . . . . .	174
5.9.2.15 m_lineStatusAll . . . . .	174
5.9.2.16 m_offsetX . . . . .	174
5.9.2.17 m_offsetY . . . . .	174
5.9.2.18 m_partSelector . . . . .	175
5.9.2.19 m_pixelDynamicRangeMax . . . . .	175
5.9.2.20 m_pixelDynamicRangeMin . . . . .	175
5.9.2.21 m_scan3dAxisMax . . . . .	175
5.9.2.22 m_scan3dAxisMin . . . . .	175
5.9.2.23 m_scan3dCoordinateOffset . . . . .	175
5.9.2.24 m_scan3dCoordinateReferenceValue . . . . .	175
5.9.2.25 m_scan3dCoordinateScale . . . . .	175
5.9.2.26 m_scan3dInvalidDataValue . . . . .	176
5.9.2.27 m_scan3dTransformValue . . . . .	176
5.9.2.28 m_scanLineSelector . . . . .	176
5.9.2.29 m_sequencerSetActive . . . . .	176



5.9.2.30 m_serialDataLength . . . . .	176
5.9.2.31 m_streamChannelID . . . . .	176
5.9.2.32 m_timerValue . . . . .	176
5.9.2.33 m_timestamp . . . . .	176
5.9.2.34 m_timestampLatchValue . . . . .	177
5.9.2.35 m_transferBlockID . . . . .	177
5.9.2.36 m_transferQueueCurrentBlockCount . . . . .	177
5.9.2.37 m_width . . . . .	177
5.10 _spinH264Option Struct Reference . . . . .	177
5.10.1 Detailed Description . . . . .	178
5.10.2 Field Documentation . . . . .	178
5.10.2.1 bitrate . . . . .	178
5.10.2.2 frameRate . . . . .	178
5.10.2.3 height . . . . .	178
5.10.2.4 reserved . . . . .	178
5.10.2.5 width . . . . .	178
5.11 _spinJPEGOption Struct Reference . . . . .	179
5.11.1 Detailed Description . . . . .	179
5.11.2 Field Documentation . . . . .	179
5.11.2.1 progressive . . . . .	179
5.11.2.2 quality . . . . .	179
5.11.2.3 reserved . . . . .	180
5.12 _spinJPG2Option Struct Reference . . . . .	180
5.12.1 Detailed Description . . . . .	180
5.12.2 Field Documentation . . . . .	180
5.12.2.1 quality . . . . .	180
5.12.2.2 reserved . . . . .	180
5.13 _spinLibraryVersion Struct Reference . . . . .	181
5.13.1 Detailed Description . . . . .	181
5.13.2 Field Documentation . . . . .	181
5.13.2.1 build . . . . .	181
5.13.2.2 major . . . . .	181
5.13.2.3 minor . . . . .	181
5.13.2.4 type . . . . .	182
5.14 _spinMJPGOption Struct Reference . . . . .	182
5.14.1 Detailed Description . . . . .	182
5.14.2 Field Documentation . . . . .	182
5.14.2.1 frameRate . . . . .	182
5.14.2.2 quality . . . . .	182
5.14.2.3 reserved . . . . .	183
5.15 _spinPGMOption Struct Reference . . . . .	183
5.15.1 Detailed Description . . . . .	183

5.15.2 Field Documentation	183
5.15.2.1 binaryFile	183
5.15.2.2 reserved	183
5.16 _spinPNGOption Struct Reference	184
5.16.1 Detailed Description	184
5.16.2 Field Documentation	184
5.16.2.1 compressionLevel	184
5.16.2.2 interlaced	184
5.16.2.3 reserved	184
5.17 _spinPPMOption Struct Reference	185
5.17.1 Detailed Description	185
5.17.2 Field Documentation	185
5.17.2.1 binaryFile	185
5.17.2.2 reserved	185
5.18 _spinTIFFOption Struct Reference	185
5.18.1 Detailed Description	186
5.18.2 Field Documentation	186
5.18.2.1 compression	186
5.18.2.2 reserved	186
<b>6 File Documentation</b>	<b>187</b>
6.1 include/spinc/CameraDefsC.h File Reference	187
6.1.1 Enumeration Type Documentation	219
6.1.1.1 _spinAcquisitionModeEnums	219
6.1.1.2 _spinAcquisitionStatusSelectorEnums	220
6.1.1.3 _spinActionUnconditionalModeEnums	220
6.1.1.4 _spinAdcBitDepthEnums	220
6.1.1.5 _spinAutoAlgorithmSelectorEnums	221
6.1.1.6 _spinAutoExposureControlPriorityEnums	221
6.1.1.7 _spinAutoExposureLightingModeEnums	221
6.1.1.8 _spinAutoExposureMeteringModeEnums	222
6.1.1.9 _spinAutoExposureTargetGreyValueAutoEnums	222
6.1.1.10 _spinBalanceRatioSelectorEnums	223
6.1.1.11 _spinBalanceWhiteAutoEnums	223
6.1.1.12 _spinBalanceWhiteAutoProfileEnums	223
6.1.1.13 _spinBinningHorizontalModeEnums	224
6.1.1.14 _spinBinningSelectorEnums	224
6.1.1.15 _spinBinningVerticalModeEnums	224
6.1.1.16 _spinBlackLevelAutoBalanceEnums	225
6.1.1.17 _spinBlackLevelAutoEnums	225
6.1.1.18 _spinBlackLevelSelectorEnums	225
6.1.1.19 _spinChunkBlackLevelSelectorEnums	226

6.1.1.20 _spinChunkCounterSelectorEnums . . . . .	226
6.1.1.21 _spinChunkEncoderSelectorEnums . . . . .	226
6.1.1.22 _spinChunkEncoderStatusEnums . . . . .	226
6.1.1.23 _spinChunkExposureTimeSelectorEnums . . . . .	227
6.1.1.24 _spinChunkGainSelectorEnums . . . . .	227
6.1.1.25 _spinChunkImageComponentEnums . . . . .	228
6.1.1.26 _spinChunkPixelFormatEnums . . . . .	228
6.1.1.27 _spinChunkRegionIDEnums . . . . .	228
6.1.1.28 _spinChunkScan3dCoordinateReferenceSelectorEnums . . . . .	229
6.1.1.29 _spinChunkScan3dCoordinateSelectorEnums . . . . .	229
6.1.1.30 _spinChunkScan3dCoordinateSystemEnums . . . . .	229
6.1.1.31 _spinChunkScan3dCoordinateSystemReferenceEnums . . . . .	230
6.1.1.32 _spinChunkScan3dCoordinateTransformSelectorEnums . . . . .	230
6.1.1.33 _spinChunkScan3dDistanceUnitEnums . . . . .	231
6.1.1.34 _spinChunkScan3dOutputModeEnums . . . . .	231
6.1.1.35 _spinChunkSelectorEnums . . . . .	232
6.1.1.36 _spinChunkSourceIDEnums . . . . .	232
6.1.1.37 _spinChunkTimerSelectorEnums . . . . .	233
6.1.1.38 _spinChunkTransferStreamIDEnums . . . . .	233
6.1.1.39 _spinCIConfigurationEnums . . . . .	233
6.1.1.40 _spinCITimeSlotsCountEnums . . . . .	234
6.1.1.41 _spinColorTransformationSelectorEnums . . . . .	234
6.1.1.42 _spinColorTransformationValueSelectorEnums . . . . .	234
6.1.1.43 _spinCounterEventActivationEnums . . . . .	235
6.1.1.44 _spinCounterEventSourceEnums . . . . .	235
6.1.1.45 _spinCounterResetActivationEnums . . . . .	236
6.1.1.46 _spinCounterResetSourceEnums . . . . .	236
6.1.1.47 _spinCounterSelectorEnums . . . . .	237
6.1.1.48 _spinCounterStatusEnums . . . . .	237
6.1.1.49 _spinCounterTriggerActivationEnums . . . . .	238
6.1.1.50 _spinCounterTriggerSourceEnums . . . . .	238
6.1.1.51 _spinCxpConnectionTestModeEnums . . . . .	239
6.1.1.52 _spinCxpLinkConfigurationEnums . . . . .	239
6.1.1.53 _spinCxpLinkConfigurationPreferredEnums . . . . .	240
6.1.1.54 _spinCxpLinkConfigurationStatusEnums . . . . .	241
6.1.1.55 _spinCxpPoCxpStatusEnums . . . . .	242
6.1.1.56 _spinDecimationHorizontalModeEnums . . . . .	242
6.1.1.57 _spinDecimationSelectorEnums . . . . .	242
6.1.1.58 _spinDecimationVerticalModeEnums . . . . .	243
6.1.1.59 _spinDefectCorrectionModeEnums . . . . .	243
6.1.1.60 _spinDeinterlacingEnums . . . . .	243
6.1.1.61 _spinDeviceCharacterSetEnums . . . . .	244

6.1.1.62 _spinDeviceClockSelectorEnums . . . . .	244
6.1.1.63 _spinDeviceConnectionStatusEnums . . . . .	244
6.1.1.64 _spinDeviceIndicatorModeEnums . . . . .	245
6.1.1.65 _spinDeviceLinkHeartbeatModeEnums . . . . .	245
6.1.1.66 _spinDeviceLinkThroughputLimitModeEnums . . . . .	245
6.1.1.67 _spinDevicePowerSupplySelectorEnums . . . . .	246
6.1.1.68 _spinDeviceRegistersEndiannessEnums . . . . .	246
6.1.1.69 _spinDeviceScanTypeEnums . . . . .	246
6.1.1.70 _spinDeviceSerialPortBaudRateEnums . . . . .	246
6.1.1.71 _spinDeviceSerialPortSelectorEnums . . . . .	247
6.1.1.72 _spinDeviceStreamChannelEndiannessEnums . . . . .	247
6.1.1.73 _spinDeviceStreamChannelTypeEnums . . . . .	247
6.1.1.74 _spinDeviceTapGeometryEnums . . . . .	248
6.1.1.75 _spinDeviceTemperatureSelectorEnums . . . . .	249
6.1.1.76 _spinDeviceTLTypeEnums . . . . .	249
6.1.1.77 _spinDeviceTypeEnums . . . . .	250
6.1.1.78 _spinEncoderModeEnums . . . . .	250
6.1.1.79 _spinEncoderOutputModeEnums . . . . .	250
6.1.1.80 _spinEncoderResetActivationEnums . . . . .	251
6.1.1.81 _spinEncoderResetSourceEnums . . . . .	251
6.1.1.82 _spinEncoderSelectorEnums . . . . .	252
6.1.1.83 _spinEncoderSourceAEnums . . . . .	253
6.1.1.84 _spinEncoderSourceBEnums . . . . .	253
6.1.1.85 _spinEncoderStatusEnums . . . . .	253
6.1.1.86 _spinEventNotificationEnums . . . . .	254
6.1.1.87 _spinEventSelectorEnums . . . . .	254
6.1.1.88 _spinExposureActiveModeEnums . . . . .	254
6.1.1.89 _spinExposureAutoEnums . . . . .	255
6.1.1.90 _spinExposureModeEnums . . . . .	255
6.1.1.91 _spinExposureTimeModeEnums . . . . .	255
6.1.1.92 _spinExposureTimeSelectorEnums . . . . .	256
6.1.1.93 _spinFileOpenModeEnums . . . . .	256
6.1.1.94 _spinFileOperationSelectorEnums . . . . .	256
6.1.1.95 _spinFileOperationStatusEnums . . . . .	257
6.1.1.96 _spinFileSelectorEnums . . . . .	257
6.1.1.97 _spinGainAutoBalanceEnums . . . . .	257
6.1.1.98 _spinGainAutoEnums . . . . .	259
6.1.1.99 _spinGainSelectorEnums . . . . .	259
6.1.1.100 _spinGevCCPEnums . . . . .	259
6.1.1.101 _spinGevCurrentPhysicalLinkConfigurationEnums . . . . .	260
6.1.1.102 _spinGevGVCPExtendedStatusCodesSelectorEnums . . . . .	260
6.1.1.103 _spinGevGVSPExtendedIDModeEnums . . . . .	260

6.1.1.104 _spinGevIEEE1588ClockAccuracyEnums . . . . .	261
6.1.1.105 _spinGevIEEE1588ModeEnums . . . . .	261
6.1.1.106 _spinGevIEEE1588StatusEnums . . . . .	261
6.1.1.107 _spinGevIPConfigurationStatusEnums . . . . .	262
6.1.1.108 _spinGevPhysicalLinkConfigurationEnums . . . . .	262
6.1.1.109 _spinGevSupportedOptionSelectorEnums . . . . .	263
6.1.1.110 _spinImageComponentSelectorEnums . . . . .	263
6.1.1.111 _spinImageCompressionJPEGFormatOptionEnums . . . . .	264
6.1.1.112 _spinImageCompressionModeEnums . . . . .	264
6.1.1.113 _spinImageCompressionRateOptionEnums . . . . .	265
6.1.1.114 _spinLineFormatEnums . . . . .	265
6.1.1.115 _spinLineInputFilterSelectorEnums . . . . .	265
6.1.1.116 _spinLineModeEnums . . . . .	266
6.1.1.117 _spinLineSelectorEnums . . . . .	266
6.1.1.118 _spinLineSourceEnums . . . . .	266
6.1.1.119 _spinLogicBlockLUTInputActivationEnums . . . . .	267
6.1.1.120 _spinLogicBlockLUTInputSelectorEnums . . . . .	267
6.1.1.121 _spinLogicBlockLUTInputSourceEnums . . . . .	268
6.1.1.122 _spinLogicBlockLUTSelectorEnums . . . . .	268
6.1.1.123 _spinLogicBlockSelectorEnums . . . . .	269
6.1.1.124 _spinLUTSelectorEnums . . . . .	269
6.1.1.125 _spinPixelColorFilterEnums . . . . .	269
6.1.1.126 _spinPixelFormatEnums . . . . .	270
6.1.1.127 _spinPixelFormatInfoSelectorEnums . . . . .	275
6.1.1.128 _spinPixelSizeEnums . . . . .	281
6.1.1.129 _spinRegionDestinationEnums . . . . .	282
6.1.1.130 _spinRegionModeEnums . . . . .	282
6.1.1.131 _spinRegionSelectorEnums . . . . .	282
6.1.1.132 _spinRgbTransformLightSourceEnums . . . . .	283
6.1.1.133 _spinScan3dCoordinateReferenceSelectorEnums . . . . .	283
6.1.1.134 _spinScan3dCoordinateSelectorEnums . . . . .	284
6.1.1.135 _spinScan3dCoordinateSystemEnums . . . . .	284
6.1.1.136 _spinScan3dCoordinateSystemReferenceEnums . . . . .	284
6.1.1.137 _spinScan3dCoordinateTransformSelectorEnums . . . . .	285
6.1.1.138 _spinScan3dDistanceUnitEnums . . . . .	285
6.1.1.139 _spinScan3dOutputModeEnums . . . . .	285
6.1.1.140 _spinSensorDigitizationTapsEnums . . . . .	286
6.1.1.141 _spinSensorShutterModeEnums . . . . .	287
6.1.1.142 _spinSensorTapsEnums . . . . .	287
6.1.1.143 _spinSequencerConfigurationModeEnums . . . . .	288
6.1.1.144 _spinSequencerConfigurationValidEnums . . . . .	288
6.1.1.145 _spinSequencerModeEnums . . . . .	288

6.1.1.146 _spinSequencerSetValidEnums . . . . .	288
6.1.1.147 _spinSequencerTriggerActivationEnums . . . . .	289
6.1.1.148 _spinSequencerTriggerSourceEnums . . . . .	289
6.1.1.149 _spinSerialPortBaudRateEnums . . . . .	289
6.1.1.150 _spinSerialPortParityEnums . . . . .	290
6.1.1.151 _spinSerialPortSelectorEnums . . . . .	290
6.1.1.152 _spinSerialPortSourceEnums . . . . .	291
6.1.1.153 _spinSerialPortStopBitsEnums . . . . .	291
6.1.1.154 _spinSoftwareSignalSelectorEnums . . . . .	291
6.1.1.155 _spinSourceSelectorEnums . . . . .	292
6.1.1.156 _spinTestPatternEnums . . . . .	292
6.1.1.157 _spinTestPatternGeneratorSelectorEnums . . . . .	292
6.1.1.158 _spinTimerSelectorEnums . . . . .	293
6.1.1.159 _spinTimerStatusEnums . . . . .	293
6.1.1.160 _spinTimerTriggerActivationEnums . . . . .	293
6.1.1.161 _spinTimerTriggerSourceEnums . . . . .	294
6.1.1.162 _spinTransferComponentSelectorEnums . . . . .	295
6.1.1.163 _spinTransferControlModeEnums . . . . .	295
6.1.1.164 _spinTransferOperationModeEnums . . . . .	296
6.1.1.165 _spinTransferQueueModeEnums . . . . .	296
6.1.1.166 _spinTransferSelectorEnums . . . . .	296
6.1.1.167 _spinTransferStatusSelectorEnums . . . . .	297
6.1.1.168 _spinTransferTriggerActivationEnums . . . . .	297
6.1.1.169 _spinTransferTriggerModeEnums . . . . .	297
6.1.1.170 _spinTransferTriggerSelectorEnums . . . . .	298
6.1.1.171 _spinTransferTriggerSourceEnums . . . . .	298
6.1.1.172 _spinTriggerActivationEnums . . . . .	299
6.1.1.173 _spinTriggerModeEnums . . . . .	300
6.1.1.174 _spinTriggerOverlapEnums . . . . .	300
6.1.1.175 _spinTriggerSelectorEnums . . . . .	300
6.1.1.176 _spinTriggerSourceEnums . . . . .	301
6.1.1.177 _spinUserOutputSelectorEnums . . . . .	301
6.1.1.178 _spinUserSetDefaultEnums . . . . .	301
6.1.1.179 _spinUserSetSelectorEnums . . . . .	302
6.1.1.180 _spinWhiteClipSelectorEnums . . . . .	302
6.2 include/spinc/ChunkDataDefC.h File Reference . . . . .	303
6.3 include/spinc/QuickSpinC.h File Reference . . . . .	304
6.3.1 Function Documentation . . . . .	304
6.3.1.1 quickSpinInit() . . . . .	304
6.3.1.2 quickSpinInitEx() . . . . .	305
6.3.1.3 quickSpinTLDeviceInit() . . . . .	305
6.3.1.4 quickSpinTLInterfacelInit() . . . . .	305

6.3.1.5 quickSpinTLStreamInit()	305
6.3.1.6 quickSpinTLSYSTEMInit()	305
6.4 include/spinc/QuickSpinDefsC.h File Reference	306
6.4.1 Typedef Documentation	306
6.4.1.1 quickSpinBooleanNode	307
6.4.1.2 quickSpinCommandNode	307
6.4.1.3 quickSpinEnumerationNode	307
6.4.1.4 quickSpinFloatNode	307
6.4.1.5 quickSpinIntegerNode	307
6.4.1.6 quickSpinRegisterNode	307
6.4.1.7 quickSpinStringNode	307
6.5 include/spinc/SpinnakerC.h File Reference	308
6.5.1 Function Documentation	316
6.5.1.1 spinCameraBeginAcquisition()	316
6.5.1.2 spinCameraDelInit()	317
6.5.1.3 spinCameraDiscoverMaxPacketSize()	317
6.5.1.4 spinCameraEndAcquisition()	318
6.5.1.5 spinCameraForceIP()	318
6.5.1.6 spinCameraGetAccessMode()	318
6.5.1.7 spinCameraGetGuiXml()	319
6.5.1.8 spinCameraGetNextImage()	319
6.5.1.9 spinCameraGetNextImageEx()	320
6.5.1.10 spinCameraGetNodeMap()	320
6.5.1.11 spinCameraGetTLDeviceNodeMap()	321
6.5.1.12 spinCameraGetTLStreamNodeMap()	321
6.5.1.13 spinCameraGetUniqueID()	322
6.5.1.14 spinCameraInit()	322
6.5.1.15 spinCamerasInitialized()	323
6.5.1.16 spinCamerasStreaming()	323
6.5.1.17 spinCamerasValid()	324
6.5.1.18 spinCameraListAppend()	324
6.5.1.19 spinCameraListClear()	325
6.5.1.20 spinCameraListCreateEmpty()	325
6.5.1.21 spinCameraListDestroy()	326
6.5.1.22 spinCameraListGet()	326
6.5.1.23 spinCameraListGetBySerial()	327
6.5.1.24 spinCameraListGetSize()	327
6.5.1.25 spinCameraListRemove()	328
6.5.1.26 spinCameraListRemoveBySerial()	328
6.5.1.27 spinCameraReadPort()	328
6.5.1.28 spinCameraRegisterDeviceEventHandler()	329
6.5.1.29 spinCameraRegisterDeviceEventHandlerEx()	329

6.5.1.30 spinCameraRegisterImageEventHandler()	330
6.5.1.31 spinCameraRelease()	330
6.5.1.32 spinCameraUnregisterDeviceEventHandler()	331
6.5.1.33 spinCameraUnregisterImageEventHandler()	331
6.5.1.34 spinCameraWritePort()	331
6.5.1.35 spinDeviceArrivalEventHandlerCreate()	332
6.5.1.36 spinDeviceArrivalEventHandlerDestroy()	332
6.5.1.37 spinDeviceEventGetId()	333
6.5.1.38 spinDeviceEventGetName()	333
6.5.1.39 spinDeviceEventGetPayloadData()	334
6.5.1.40 spinDeviceEventGetPayloadDataSize()	334
6.5.1.41 spinDeviceEventHandlerCreate()	335
6.5.1.42 spinDeviceEventHandlerDestroy()	335
6.5.1.43 spinDeviceRemovalEventHandlerCreate()	336
6.5.1.44 spinDeviceRemovalEventHandlerDestroy()	336
6.5.1.45 spinErrorGetLast()	337
6.5.1.46 spinErrorGetLastBuildDate()	337
6.5.1.47 spinErrorGetLastBuildTime()	338
6.5.1.48 spinErrorGetLastFileName()	338
6.5.1.49 spinErrorGetLastFullMessage()	339
6.5.1.50 spinErrorGetLastFunctionName()	339
6.5.1.51 spinErrorGetLastLineNumber()	340
6.5.1.52 spinErrorGetLastMessage()	340
6.5.1.53 spinImageCalculateStatistics()	341
6.5.1.54 spinImageCheckCRC()	341
6.5.1.55 spinImageChunkDataGetFloatValue()	341
6.5.1.56 spinImageChunkDataGetIntValue()	342
6.5.1.57 spinImageConvert()	342
6.5.1.58 spinImageConvertEx()	342
6.5.1.59 spinImageCreate()	343
6.5.1.60 spinImageCreateEmpty()	343
6.5.1.61 spinImageCreateEx()	344
6.5.1.62 spinImageDeepCopy()	344
6.5.1.63 spinImageDestroy()	345
6.5.1.64 spinImageEventHandlerCreate()	345
6.5.1.65 spinImageEventHandlerDestroy()	346
6.5.1.66 spinImageGetBitsPerPixel()	346
6.5.1.67 spinImageGetBufferSize()	347
6.5.1.68 spinImageGetChunkLayoutID()	347
6.5.1.69 spinImageGetColorProcessing()	348
6.5.1.70 spinImageGetData()	348
6.5.1.71 spinImageGetDefaultColorProcessing()	349



6.5.1.72 spinImageGetFrameID()	349
6.5.1.73 spinImageGetHeight()	350
6.5.1.74 spinImageGetID()	350
6.5.1.75 spinImageGetOffsetX()	351
6.5.1.76 spinImageGetOffsetY()	351
6.5.1.77 spinImageGetPaddingX()	352
6.5.1.78 spinImageGetPaddingY()	352
6.5.1.79 spinImageGetPayloadType()	353
6.5.1.80 spinImageGetPixelFormat()	353
6.5.1.81 spinImageGetPixelFormatName()	354
6.5.1.82 spinImageGetPrivateData()	354
6.5.1.83 spinImageGetSize()	355
6.5.1.84 spinImageGetStatus()	355
6.5.1.85 spinImageGetStatusDescription()	356
6.5.1.86 spinImageGetStride()	356
6.5.1.87 spinImageGetTimeStamp()	357
6.5.1.88 spinImageGetTLPayloadType()	357
6.5.1.89 spinImageGetTLPixelFormat()	358
6.5.1.90 spinImageGetTLPixelFormatNamespace()	358
6.5.1.91 spinImageGetValidPayloadSize()	359
6.5.1.92 spinImageGetWidth()	359
6.5.1.93 spinImageHasCRC()	360
6.5.1.94 spinImageIsIncomplete()	360
6.5.1.95 spinImageRelease()	361
6.5.1.96 spinImageReset()	361
6.5.1.97 spinImageResetEx()	362
6.5.1.98 spinImageSave()	362
6.5.1.99 spinImageSaveBmp()	363
6.5.1.100 spinImageSaveFromExt()	363
6.5.1.101 spinImageSaveJpeg()	364
6.5.1.102 spinImageSaveJpg2()	364
6.5.1.103 spinImageSavePgm()	365
6.5.1.104 spinImageSavePng()	365
6.5.1.105 spinImageSavePpm()	366
6.5.1.106 spinImageSaveTiff()	366
6.5.1.107 spinImageSetDefaultColorProcessing()	367
6.5.1.108 spinImageStatisticsCreate()	367
6.5.1.109 spinImageStatisticsDestroy()	368
6.5.1.110 spinImageStatisticsDisableAll()	368
6.5.1.111 spinImageStatisticsEnableAll()	368
6.5.1.112 spinImageStatisticsEnableGreyOnly()	369
6.5.1.113 spinImageStatisticsEnableHslOnly()	369

6.5.1.114 spinImageStatisticsEnableRgbOnly()	370
6.5.1.115 spinImageStatisticsGetAll()	370
6.5.1.116 spinImageStatisticsGetChannelStatus()	371
6.5.1.117 spinImageStatisticsGetHistogram()	371
6.5.1.118 spinImageStatisticsGetMean()	372
6.5.1.119 spinImageStatisticsGetNumPixelValues()	372
6.5.1.120 spinImageStatisticsGetPixelValueRange()	373
6.5.1.121 spinImageStatisticsGetRange()	373
6.5.1.122 spinImageStatisticsSetChannelStatus()	374
6.5.1.123 spinInterfaceEventHandlerCreate()	374
6.5.1.124 spinInterfaceEventHandlerDestroy()	375
6.5.1.125 spinInterfaceGetCameras()	375
6.5.1.126 spinInterfaceGetCamerasEx()	376
6.5.1.127 spinInterfaceGetTLNodeMap()	376
6.5.1.128 spinInterfaceIsInUse()	377
6.5.1.129 spinInterfaceListClear()	377
6.5.1.130 spinInterfaceListCreateEmpty()	378
6.5.1.131 spinInterfaceListDestroy()	378
6.5.1.132 spinInterfaceListGet()	379
6.5.1.133 spinInterfaceListGetSize()	379
6.5.1.134 spinInterfaceRegisterDeviceArrivalEventHandler()	380
6.5.1.135 spinInterfaceRegisterDeviceRemovalEventHandler()	380
6.5.1.136 spinInterfaceRegisterInterfaceEventHandler()	381
6.5.1.137 spinInterfaceRelease()	381
6.5.1.138 spinInterfaceSendActionCommand()	382
6.5.1.139 spinInterfaceUnregisterDeviceArrivalEventHandler()	382
6.5.1.140 spinInterfaceUnregisterDeviceRemovalEventHandler()	383
6.5.1.141 spinInterfaceUnregisterInterfaceEventHandler()	383
6.5.1.142 spinInterfaceUpdateCameras()	384
6.5.1.143 spinLogDataGetCategoryName()	384
6.5.1.144 spinLogDataGetLogMessage()	385
6.5.1.145 spinLogDataGetNDC()	385
6.5.1.146 spinLogDataGetPriority()	386
6.5.1.147 spinLogDataGetPriorityName()	386
6.5.1.148 spinLogDataGetThreadName()	387
6.5.1.149 spinLogDataGetTimestamp()	387
6.5.1.150 spinLogEventHandlerCreate()	388
6.5.1.151 spinLogEventHandlerDestroy()	388
6.5.1.152 spinSystemGetCameras()	389
6.5.1.153 spinSystemGetCamerasEx()	389
6.5.1.154 spinSystemGetInstance()	390
6.5.1.155 spinSystemGetInterfaces()	390

6.5.1.156 spinSystemGetLibraryVersion()	391
6.5.1.157 spinSystemGetLoggingLevel()	391
6.5.1.158 spinSystemGetTLNodeMap()	392
6.5.1.159 spinSystemIsInUse()	392
6.5.1.160 spinSystemRegisterDeviceArrivalEventHandler()	392
6.5.1.161 spinSystemRegisterDeviceRemovalEventHandler()	393
6.5.1.162 spinSystemRegisterInterfaceEventHandler()	393
6.5.1.163 spinSystemRegisterLogEventHandler()	394
6.5.1.164 spinSystemReleaseInstance()	394
6.5.1.165 spinSystemSendActionCommand()	395
6.5.1.166 spinSystemSetLoggingLevel()	396
6.5.1.167 spinSystemUnregisterAllLogEventHandlers()	396
6.5.1.168 spinSystemUnregisterDeviceArrivalEventHandler()	397
6.5.1.169 spinSystemUnregisterDeviceRemovalEventHandler()	397
6.5.1.170 spinSystemUnregisterInterfaceEventHandler()	398
6.5.1.171 spinSystemUnregisterLogEventHandler()	398
6.5.1.172 spinSystemUpdateCameras()	399
6.5.1.173 spinSystemUpdateCamerasEx()	399
6.6 include/spinc/SpinnakerDefsC.h File Reference	400
6.6.1 Typedef Documentation	404
6.6.1.1 bool8_t	404
6.6.1.2 spinArrivalEventFunction	405
6.6.1.3 spinCamera	405
6.6.1.4 spinCameraList	405
6.6.1.5 spinDeviceArrivalEventHandler	405
6.6.1.6 spinDeviceEventData	405
6.6.1.7 spinDeviceEventFunction	405
6.6.1.8 spinDeviceEventHandler	406
6.6.1.9 spinDeviceRemovalEventHandler	406
6.6.1.10 spinImage	406
6.6.1.11 spinImageEventFunction	406
6.6.1.12 spinImageEventHandler	406
6.6.1.13 spinImageStatistics	406
6.6.1.14 spinInterface	407
6.6.1.15 spinInterfaceEventHandler	407
6.6.1.16 spinInterfaceList	407
6.6.1.17 spinLogEventData	407
6.6.1.18 spinLogEventFunction	407
6.6.1.19 spinLogEventHandler	407
6.6.1.20 spinRemovalEventFunction	408
6.6.1.21 spinSystem	408
6.6.1.22 spinVideo	408

6.6.2 Enumeration Type Documentation	408
6.6.2.1 _actionCommandStatus	408
6.6.2.2 _spinColorProcessingAlgorithm	408
6.6.2.3 _spinError	409
6.6.2.4 _spinImageFileFormat	410
6.6.2.5 _spinImageStatus	411
6.6.2.6 _spinLogLevel	411
6.6.2.7 _spinPayloadTypeInfolDs	412
6.6.2.8 _spinPixelFormatNamespaceID	412
6.6.2.9 _spinStatisticsChannel	413
6.6.2.10 CompressionMethod	413
6.6.3 Variable Documentation	414
6.6.3.1 False	414
6.6.3.2 True	414
6.7 include/spinc/SpinnakerGenApiC.h File Reference	414
6.7.1 Function Documentation	418
6.7.1.1 spinBooleanGetValue()	418
6.7.1.2 spinBooleanSetValue()	419
6.7.1.3 spinCategoryGetFeatureByIndex()	419
6.7.1.4 spinCategoryGetNumFeatures()	420
6.7.1.5 spinCommandExecute()	420
6.7.1.6 spinCommandIsDone()	421
6.7.1.7 spinEnumerationEntryGetEnumValue()	421
6.7.1.8 spinEnumerationEntryGetIntValue()	422
6.7.1.9 spinEnumerationEntryGetSymbolic()	422
6.7.1.10 spinEnumerationGetCurrentEntry()	423
6.7.1.11 spinEnumerationGetEntryByIndex()	423
6.7.1.12 spinEnumerationGetEntryByName()	424
6.7.1.13 spinEnumerationGetNumEntries()	424
6.7.1.14 spinEnumerationSetEnumValue()	425
6.7.1.15 spinEnumerationSetIntValue()	425
6.7.1.16 spinFloatGetMax()	426
6.7.1.17 spinFloatGetMin()	426
6.7.1.18 spinFloatGetRepresentation()	426
6.7.1.19 spinFloatGetUnit()	427
6.7.1.20 spinFloatGetValue()	427
6.7.1.21 spinFloatGetValueEx()	428
6.7.1.22 spinFloatSetValue()	428
6.7.1.23 spinFloatSetValueEx()	429
6.7.1.24 spinIntegerGetInc()	429
6.7.1.25 spinIntegerGetMax()	430
6.7.1.26 spinIntegerGetMin()	430

---

6.7.1.27 spinIntegerGetRepresentation()	431
6.7.1.28 spinIntegerGetValue()	431
6.7.1.29 spinIntegerGetValueEx()	432
6.7.1.30 spinIntegerSetValue()	432
6.7.1.31 spinIntegerSetValueEx()	433
6.7.1.32 spinNodeDeregisterCallback()	433
6.7.1.33 spinNodeFromString()	434
6.7.1.34 spinNodeFromStringEx()	434
6.7.1.35 spinNodeGetAccessMode()	435
6.7.1.36 spinNodeGetCachingMode()	435
6.7.1.37 spinNodeGetDescription()	436
6.7.1.38 spinNodeGetDisplayName()	436
6.7.1.39 spinNodeGetImposedAccessMode()	437
6.7.1.40 spinNodeGetImposedVisibility()	437
6.7.1.41 spinNodeGetName()	438
6.7.1.42 spinNodeGetNameSpace()	438
6.7.1.43 spinNodeGetPollingTime()	439
6.7.1.44 spinNodeGetToolTip()	439
6.7.1.45 spinNodeGetType()	440
6.7.1.46 spinNodeGetVisibility()	440
6.7.1.47 spinNodeInvalidateNode()	441
6.7.1.48 spinNodesAvailable()	441
6.7.1.49 spinNodesEqual()	442
6.7.1.50 spinNodesImplemented()	442
6.7.1.51 spinNodesReadable()	443
6.7.1.52 spinNodesWritable()	443
6.7.1.53 spinNodeMapGetNode()	444
6.7.1.54 spinNodeMapGetNodeByIndex()	444
6.7.1.55 spinNodeMapGetNumNodes()	445
6.7.1.56 spinNodeMapPoll()	445
6.7.1.57 spinNodeRegisterCallback()	446
6.7.1.58 spinNodeToString()	446
6.7.1.59 spinNodeToStringEx()	447
6.7.1.60 spinRegisterGet()	447
6.7.1.61 spinRegisterGetAddress()	448
6.7.1.62 spinRegisterGetEx()	448
6.7.1.63 spinRegisterGetLength()	449
6.7.1.64 spinRegisterSet()	449
6.7.1.65 spinRegisterSetEx()	450
6.7.1.66 spinRegisterSetReference()	450
6.7.1.67 spinStringGetMaxLength()	451
6.7.1.68 spinStringGetValue()	451

6.7.1.69 spinStringGetValueEx()	452
6.7.1.70 spinStringSetValue()	452
6.7.1.71 spinStringSetValueEx()	453
6.8 include/spinc/SpinnakerGenApiDefsC.h File Reference	453
6.8.1 Typedef Documentation	456
6.8.1.1 spinNodeCallbackFunction	456
6.8.1.2 spinNodeCallbackHandle	456
6.8.1.3 spinNodeHandle	456
6.8.1.4 spinNodeMapHandle	457
6.8.2 Enumeration Type Documentation	457
6.8.2.1 _spinAccessMode	457
6.8.2.2 _spinCachingMode	457
6.8.2.3 _spinDisplayNotation	457
6.8.2.4 _spinEndianess	458
6.8.2.5 _spinIncMode	458
6.8.2.6 _spinInputDirection	458
6.8.2.7 _spinInterfaceType	459
6.8.2.8 _spinLinkType	460
6.8.2.9 _spinNameSpace	460
6.8.2.10 _spinNodeType	461
6.8.2.11 _spinRepresentation	461
6.8.2.12 _spinSign	461
6.8.2.13 _spinSlope	462
6.8.2.14 _spinStandardNameSpace	462
6.8.2.15 _spinVisibility	463
6.8.2.16 _spinXMLValidation	463
6.8.2.17 _spinYesNo	463
6.9 include/spinc/SpinnakerPlatformC.h File Reference	464
6.9.1 Macro Definition Documentation	464
6.9.1.1 SPINNAKERC_API	465
6.10 include/spinc/SpinVideoC.h File Reference	465
6.10.1 Function Documentation	465
6.10.1.1 spinVideoAppend()	466
6.10.1.2 spinVideoClose()	466
6.10.1.3 spinVideoOpenH264()	466
6.10.1.4 spinVideoOpenMJPEG()	466
6.10.1.5 spinVideoOpenUncompressed()	466
6.10.1.6 spinVideoSetMaximumFileSize()	466
6.11 include/spinc/TransportLayerDefsC.h File Reference	467
6.11.1 Enumeration Type Documentation	469
6.11.1.1 _spinTLDeviceAccessStatusEnums	469
6.11.1.2 _spinTLDeviceCurrentSpeedEnums	469

6.11.1.3 _spinTLDeviceEndianessMechanismEnums . . . . .	469
6.11.1.4 _spinTLDeviceTypeEnums . . . . .	471
6.11.1.5 _spinTLFilterDriverStatusEnums . . . . .	471
6.11.1.6 _spinTLGenICamXMLLocationEnums . . . . .	471
6.11.1.7 _spinTLGevCCPEnums . . . . .	472
6.11.1.8 _spinTLGUIXMLLocationEnums . . . . .	472
6.11.1.9 _spinTLInterfaceTypeEnums . . . . .	472
6.11.1.10 _spinTLPOEStatusEnums . . . . .	473
6.11.1.11 _spinTLStreamBufferCountModeEnums . . . . .	473
6.11.1.12 _spinTLStreamBufferHandlingModeEnums . . . . .	473
6.11.1.13 _spinTLStreamTypeEnums . . . . .	474
6.11.1.14 _spinTLTLTypeEnums . . . . .	475
6.12 include/spinc/TransportLayerDeviceC.h File Reference . . . . .	475
6.13 include/spinc/TransportLayerInterfaceC.h File Reference . . . . .	476
6.14 include/spinc/TransportLayerStreamC.h File Reference . . . . .	476
6.15 include/spinc/TransportLayerSystemC.h File Reference . . . . .	477
<b>Index</b>	<b>479</b>





# Chapter 1

## Module Index

### 1.1 Modules

Here is a list of all modules:

Camera Enumerations . . . . .	8
Chunk Data Structures . . . . .	9
Spinnaker C QuickSpin API . . . . .	10
TLDevice Structures . . . . .	46
TLInterface Structures . . . . .	47
TLStream Structures . . . . .	48
TLSystem Structures . . . . .	49
QuickSpin Access . . . . .	11
Spinnaker C API . . . . .	12
Spinnaker C Definitions . . . . .	7
Error Handling . . . . .	13
System Access . . . . .	14
InterfaceList Access . . . . .	15
CameraList Access . . . . .	16
Interface Access . . . . .	17
Camera Access . . . . .	18
Image Access . . . . .	19
Event Access . . . . .	20
ImageStatistics Access . . . . .	21
Logging Event Data Access . . . . .	22
Device Event Data Access . . . . .	23
Chunk data access . . . . .	24
Spinnaker C Handles . . . . .	25
Spinnaker C Function Signatures . . . . .	26
Spinnaker C Enumerations . . . . .	27
Spinnaker C Structures . . . . .	28
Spinnaker C GenICam API . . . . .	29
Node Map Access . . . . .	30
Node Access . . . . .	31
IValue Access . . . . .	32
String Access . . . . .	33
Integer Access . . . . .	34
IFloat Access . . . . .	35
IEnumeration Access . . . . .	36
IEnumEntry Access . . . . .	37

IBoolean Access . . . . .	38
ICommand Access . . . . .	39
ICategory Access . . . . .	40
IRegister Access . . . . .	41
Spinnaker C GenICam Handles . . . . .	42
Spinnaker C GenICam Enumerations . . . . .	43
SpinVideo Recording Access . . . . .	44
Transport Layer Enumerations . . . . .	45

## Chapter 2

# Data Structure Index

### 2.1 Data Structures

Here are the data structures with brief descriptions:

<a href="#">_actionCommandResult</a>	
Action Command Result . . . . .	51
<a href="#">_quickSpin</a> . . . . .	52
<a href="#">_quickSpinTLDevice</a> . . . . .	148
<a href="#">_quickSpinTLInterface</a> . . . . .	154
<a href="#">_quickSpinTLStream</a> . . . . .	162
<a href="#">_quickSpinTLSystem</a> . . . . .	166
<a href="#">_spinAVIOption</a>	
Options for saving uncompressed videos . . . . .	170
<a href="#">_spinBMPOption</a>	
Options for saving BMP images . . . . .	170
<a href="#">_spinChunkData</a>	
The type of information that can be obtained from image chunk data . . . . .	171
<a href="#">_spinH264Option</a>	
Options for saving H264 videos . . . . .	177
<a href="#">_spinJPEGOption</a>	
Options for saving JPEG images . . . . .	179
<a href="#">_spinJPG2Option</a>	
Options for saving JPEG 2000 images . . . . .	180
<a href="#">_spinLibraryVersion</a>	
Provides easier access to the current version of Spinnaker . . . . .	181
<a href="#">_spinMJPGOption</a>	
Options for saving MJPG videos . . . . .	182
<a href="#">_spinPGMOption</a>	
Options for saving PGM images . . . . .	183
<a href="#">_spinPNGOption</a>	
Options for saving PNG images . . . . .	184
<a href="#">_spinPPMOption</a>	
Options for saving PPM images . . . . .	185
<a href="#">_spinTIFFOption</a>	
Options for saving TIFF images . . . . .	185



## Chapter 3

# File Index

### 3.1 File List

Here is a list of all files with brief descriptions:

include/spinc/ <a href="#">CameraDefsC.h</a>	187
include/spinc/ <a href="#">ChunkDataDefC.h</a>	303
include/spinc/ <a href="#">QuickSpinC.h</a>	304
include/spinc/ <a href="#">QuickSpinDefsC.h</a>	306
include/spinc/ <a href="#">SpinnakerC.h</a>	308
include/spinc/ <a href="#">SpinnakerDefsC.h</a>	400
include/spinc/ <a href="#">SpinnakerGenApiC.h</a>	414
include/spinc/ <a href="#">SpinnakerGenApiDefsC.h</a>	453
include/spinc/ <a href="#">SpinnakerPlatformC.h</a>	464
include/spinc/ <a href="#">SpinVideoC.h</a>	465
include/spinc/ <a href="#">TransportLayerDefsC.h</a>	467
include/spinc/ <a href="#">TransportLayerDeviceC.h</a>	475
include/spinc/ <a href="#">TransportLayerInterfaceC.h</a>	476
include/spinc/ <a href="#">TransportLayerStreamC.h</a>	476
include/spinc/ <a href="#">TransportLayerSystemC.h</a>	477



## Chapter 4

# Module Documentation

### 4.1 Spinnaker C Definitions

Collaboration diagram for Spinnaker C Definitions:



Definitions for Spinnaker C

Definitions for Spinnaker C API

Holds enumerations, typedefs and structures that are used across the Spinnaker C API wrapper.

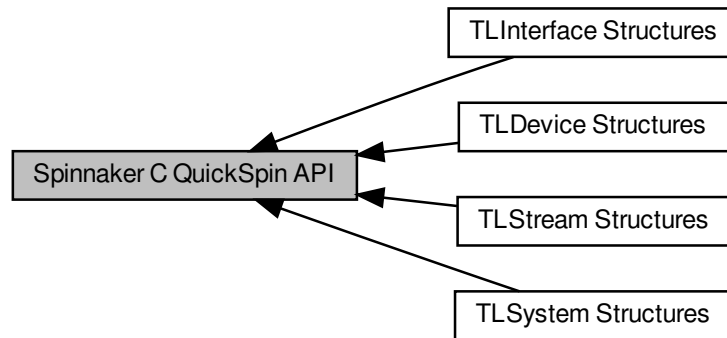
## 4.2 Camera Enumerations



## 4.3 Chunk Data Structures

## 4.4 Spinnaker C QuickSpin API

Collaboration diagram for Spinnaker C QuickSpin API:



### Modules

- [TLDevice Structures](#)
- [TLInterface Structures](#)
- [TLStream Structures](#)
- [TLSystem Structures](#)

#### 4.4.1 Detailed Description

## 4.5 QuickSpin Access

The functions in this section initialize the various QuickSpin structs for the C API.

The functions in this section initialize the various QuickSpin structs for the C API.

## 4.6 Spinnaker C API

SpinnakerPlatform C Include.

Collaboration diagram for Spinnaker C API:



### Modules

- [Spinnaker C Definitions](#)

#### 4.6.1 Detailed Description

SpinnakerPlatform C Include.

Spinnaker C Definition Includes Spinnaker GenICam C Wrapper Includes Spinnaker QuickSpin C Includes

Spinnaker C Definition Includes

## 4.7 Error Handling

The functions in this section provide access to additional information related to error returns.

The functions in this section provide access to additional information related to error returns.

## 4.8 System Access

The functions in this section provide access to information, objects, and functionality of the system object. This includes the system object, interface and camera lists, and interface and logging events.

The functions in this section provide access to information, objects, and functionality of the system object. This includes the system object, interface and camera lists, and interface and logging events.

## 4.9 InterfaceList Access

The functions in this section provide access to information, objects, and functionality of interface lists. This includes updating, size and interface retrieval, and clearance.

The functions in this section provide access to information, objects, and functionality of interface lists. This includes updating, size and interface retrieval, and clearance.

## 4.10 CameraList Access

The functions in this section provide access to information, objects, and functionality of camera lists. This includes updating, size and camera retrieval, and clearance.

The functions in this section provide access to information, objects, and functionality of camera lists. This includes updating, size and camera retrieval, and clearance.



## 4.11 Interface Access

The functions in this section provide access to information, objects, and functionality of interfaces. This includes camera list and nodemap retrieval, event handler registration, and interface release.

The functions in this section provide access to information, objects, and functionality of interfaces. This includes camera list and nodemap retrieval, event handler registration, and interface release.

## 4.12 Camera Access

The functions in this section provide access to information, objects, and functionality of cameras. This includes nodemap retrieval, acquisition and init commands, event handler registration, and camera property retrieval.

The functions in this section provide access to information, objects, and functionality of cameras. This includes nodemap retrieval, acquisition and init commands, event handler registration, and camera property retrieval.

## 4.13 Image Access

The functions in this section provide access to information and functionality of images. This includes creation, destruction, and saving as well as a wealth of information including things like width, height, stride, and timestamp.

The functions in this section provide access to information and functionality of images. This includes creation, destruction, and saving as well as a wealth of information including things like width, height, stride, and timestamp.

## 4.14 Event Access

The functions in this section allow for the creation and destruction of events.

The functions in this section allow for the creation and destruction of events.

## 4.15 ImageStatistics Access

The functions in this section provide access to information and functionality related to image statistics. This includes context creation and destruction, the enabling and disabling of channels, and value retrieval.

The functions in this section provide access to information and functionality related to image statistics. This includes context creation and destruction, the enabling and disabling of channels, and value retrieval.

## 4.16 Logging Event Data Access

The functions in this section allow for the retrieval of logging event data.

The functions in this section allow for the retrieval of logging event data.

## 4.17 Device Event Data Access

The functions in this section allow for the retrieval of device event data.

The functions in this section allow for the retrieval of device event data.

## 4.18 Chunk data access

The functions in this section provide access to chunk data stored on images.

The functions in this section provide access to chunk data stored on images.



## 4.19 Spinnaker C Handles

Spinnaker C handle definitions.

Spinnaker C handle definitions.

## 4.20 Spinnaker C Function Signatures

Spinnaker C function signature definitions.

Spinnaker C function signature definitions.

## 4.21 Spinnaker C Enumerations

Spinnaker C enumeration definitions.

Spinnaker C enumeration definitions.

## 4.22 Spinnaker C Structures

Spinnaker C structure definitions.

Spinnaker C structure definitions.

## 4.23 Spinnaker C GenICam API

## 4.24 Node Map Access

The functions in this section provide access to information, objects, and functionality related to nodemaps. This includes nodes, node counts, and polling.

The functions in this section provide access to information, objects, and functionality related to nodemaps. This includes nodes, node counts, and polling.

## 4.25 Node Access

The functions in this section provide access to information and objects retrieved from nodes. This includes node properties and callback registration.

The functions in this section provide access to information and objects retrieved from nodes. This includes node properties and callback registration.

## 4.26 IValue Access

The functions in this section provide access to nodes as value nodes. As value nodes are not an actual node type, the functions are named as regular nodes. Functions include reading from and writing to any node with a string.

The functions in this section provide access to nodes as value nodes. As value nodes are not an actual node type, the functions are named as regular nodes. Functions include reading from and writing to any node with a string.



## 4.27 String Access

The functions in this section provide access to string nodes using character pointers and arrays. This includes getters and setters of values and value lengths.

The functions in this section provide access to string nodes using character pointers and arrays. This includes getters and setters of values and value lengths.

## 4.28 Integer Access

The functions in this section provide access to integer nodes using the `int64_t` data type. This includes value getters and setters, min, max, and increment functions, and node representation.

The functions in this section provide access to integer nodes using the `int64_t` data type. This includes value getters and setters, min, max, and increment functions, and node representation.

## 4.29 IFloat Access

The functions in this section provide access to float nodes using double as the data type. This includes value getters and setters, min and max functions, and node representation.

The functions in this section provide access to float nodes using double as the data type. This includes value getters and setters, min and max functions, and node representation.

## 4.30 IEnumeration Access

The functions in this section provide access to enum nodes. This includes retrieving the number of entries, an entry by index or name, retrieving the current entry node, or setting the node using an integer.

The functions in this section provide access to enum nodes. This includes retrieving the number of entries, an entry by index or name, retrieving the current entry node, or setting the node using an integer.

## 4.31 IEnumEntry Access

The functions in this section provide access to entry nodes. This includes retrieving the integer value or the symbolic of an entry.

The functions in this section provide access to entry nodes. This includes retrieving the integer value or the symbolic of an entry.

## 4.32 IBoolean Access

The functions in this section provide access to boolean nodes using the `bool8_t` data type, values represented with 'True' and 'False'. This includes value getters and setters.

The functions in this section provide access to boolean nodes using the `bool8_t` data type, values represented with 'True' and 'False'. This includes value getters and setters.

## 4.33 ICommand Access

The functions in this section all provide access to information and objects retrieved from nodes. This includes node properties and callbacks.

The functions in this section all provide access to information and objects retrieved from nodes. This includes node properties and callbacks.

## 4.34 ICategory Access

The functions in this section all provide access to information and objects retrieved from nodes. This includes node properties and callbacks.

The functions in this section all provide access to information and objects retrieved from nodes. This includes node properties and callbacks.



## 4.35 IRegister Access

The functions in this section provide access to register nodes. This includes access to the node, its address and length, and reference.

The functions in this section provide access to register nodes. This includes access to the node, its address and length, and reference.

## 4.36 Spinnaker C GenICam Handles

Handle definitions for Spinnaker C GenICam API.

Handle definitions for Spinnaker C GenICam API.

## 4.37 Spinnaker C GenICam Enumerations

Enumeration definitions for Spinnaker C GenICam API.

Enumeration definitions for Spinnaker C GenICam API.

## 4.38 SpinVideo Recording Access

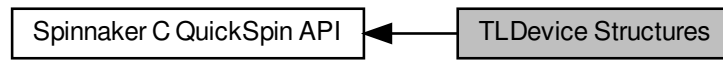
The functions in this section provide access to video recording capabilities, which include opening, building, and closing video files.

The functions in this section provide access to video recording capabilities, which include opening, building, and closing video files.

## 4.39 Transport Layer Enumerations

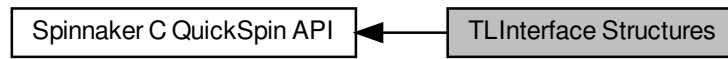
## 4.40 TLDevice Structures

Collaboration diagram for TLDevice Structures:



## 4.41 TLInterface Structures

Collaboration diagram for TLInterface Structures:



## 4.42 TLStream Structures

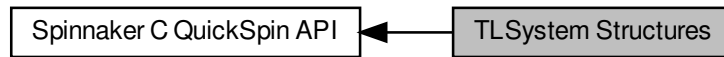
Collaboration diagram for TLStream Structures:





## 4.43 TLSystem Structures

Collaboration diagram for TLSystem Structures:





## Chapter 5

# Data Structure Documentation

### 5.1 `_actionCommandResult` Struct Reference

Action Command Result.

#### Data Fields

- unsigned int [DeviceAddress](#)
- actionCommandStatus [Status](#)

#### 5.1.1 Detailed Description

Action Command Result.

#### 5.1.2 Field Documentation

##### 5.1.2.1 DeviceAddress

```
unsigned int DeviceAddress
```

##### 5.1.2.2 Status

```
actionCommandStatus Status
```

The documentation for this struct was generated from the following file:

- include/spinc/[SpinnakerDefsC.h](#)

## 5.2 `_quickSpin` Struct Reference

### Data Fields

- `quickSpinIntegerNode` LUTIndex
- `quickSpinBooleanNode` LUTEnable
- `quickSpinIntegerNode` LUTValue
- `quickSpinEnumerationNode` LUTSelector
- `quickSpinFloatNode` ExposureTime
- `quickSpinCommandNode` AcquisitionStop
- `quickSpinFloatNode` AcquisitionResultingFrameRate
- `quickSpinFloatNode` AcquisitionLineRate
- `quickSpinCommandNode` AcquisitionStart
- `quickSpinCommandNode` TriggerSoftware
- `quickSpinEnumerationNode` ExposureMode
- `quickSpinEnumerationNode` AcquisitionMode
- `quickSpinIntegerNode` AcquisitionFrameCount
- `quickSpinEnumerationNode` TriggerSource
- `quickSpinEnumerationNode` TriggerActivation
- `quickSpinEnumerationNode` SensorShutterMode
- `quickSpinFloatNode` TriggerDelay
- `quickSpinEnumerationNode` TriggerMode
- `quickSpinFloatNode` AcquisitionFrameRate
- `quickSpinEnumerationNode` TriggerOverlap
- `quickSpinEnumerationNode` TriggerSelector
- `quickSpinBooleanNode` AcquisitionFrameRateEnable
- `quickSpinEnumerationNode` ExposureAuto
- `quickSpinIntegerNode` AcquisitionBurstFrameCount
- `quickSpinIntegerNode` EventTest
- `quickSpinIntegerNode` EventTestTimestamp
- `quickSpinIntegerNode` EventExposureEndFrameID
- `quickSpinIntegerNode` EventExposureEnd
- `quickSpinIntegerNode` EventExposureEndTimestamp
- `quickSpinIntegerNode` EventError
- `quickSpinIntegerNode` EventErrorTimestamp
- `quickSpinIntegerNode` EventErrorCode
- `quickSpinIntegerNode` EventErrorFrameID
- `quickSpinEnumerationNode` EventSelector
- `quickSpinBooleanNode` EventSerialReceiveOverflow
- `quickSpinIntegerNode` EventSerialPortReceive
- `quickSpinIntegerNode` EventSerialPortReceiveTimestamp
- `quickSpinStringNode` EventSerialData
- `quickSpinIntegerNode` EventSerialDataLength
- `quickSpinEnumerationNode` EventNotification
- `quickSpinIntegerNode` LogicBlockLUTRowIndex
- `quickSpinEnumerationNode` LogicBlockSelector
- `quickSpinEnumerationNode` LogicBlockLUTInputActivation
- `quickSpinEnumerationNode` LogicBlockLUTInputSelector
- `quickSpinEnumerationNode` LogicBlockLUTInputSource
- `quickSpinBooleanNode` LogicBlockLUTOutputValue
- `quickSpinIntegerNode` LogicBlockLUTOutputValueAll
- `quickSpinEnumerationNode` LogicBlockLUTSelector
- `quickSpinFloatNode` ColorTransformationValue
- `quickSpinBooleanNode` ColorTransformationEnable

- quickSpinEnumerationNode ColorTransformationSelector
- quickSpinEnumerationNode RgbTransformLightSource
- quickSpinFloatNode Saturation
- quickSpinBooleanNode SaturationEnable
- quickSpinEnumerationNode ColorTransformationValueSelector
- quickSpinIntegerNode TimestampLatchValue
- quickSpinCommandNode TimestampReset
- quickSpinStringNode DeviceUserID
- quickSpinFloatNode DeviceTemperature
- quickSpinIntegerNode MaxDeviceResetTime
- quickSpinIntegerNode DeviceTLVersionMinor
- quickSpinStringNode DeviceSerialNumber
- quickSpinStringNode DeviceVendorName
- quickSpinEnumerationNode DeviceRegistersEndianness
- quickSpinStringNode DeviceManufacturerInfo
- quickSpinIntegerNode DeviceLinkSpeed
- quickSpinIntegerNode LinkUptime
- quickSpinIntegerNode DeviceEventChannelCount
- quickSpinCommandNode TimestampLatch
- quickSpinEnumerationNode DeviceScanType
- quickSpinCommandNode DeviceReset
- quickSpinEnumerationNode DeviceCharacterSet
- quickSpinIntegerNode DeviceLinkThroughputLimit
- quickSpinStringNode DeviceFirmwareVersion
- quickSpinIntegerNode DeviceStreamChannelCount
- quickSpinEnumerationNode DeviceTLType
- quickSpinStringNode DeviceVersion
- quickSpinEnumerationNode DevicePowerSupplySelector
- quickSpinStringNode SensorDescription
- quickSpinStringNode DeviceModelName
- quickSpinIntegerNode DeviceTLVersionMajor
- quickSpinEnumerationNode DeviceTemperatureSelector
- quickSpinIntegerNode EnumerationCount
- quickSpinFloatNode PowerSupplyCurrent
- quickSpinStringNode DeviceID
- quickSpinIntegerNode DeviceUptime
- quickSpinIntegerNode DeviceLinkCurrentThroughput
- quickSpinIntegerNode DeviceMaxThroughput
- quickSpinCommandNode FactoryReset
- quickSpinFloatNode PowerSupplyVoltage
- quickSpinEnumerationNode DeviceIndicatorMode
- quickSpinFloatNode DeviceLinkBandwidthReserve
- quickSpinIntegerNode AasRoiOffsetY
- quickSpinIntegerNode AasRoiOffsetX
- quickSpinEnumerationNode AutoExposureControlPriority
- quickSpinFloatNode BalanceWhiteAutoLowerLimit
- quickSpinFloatNode BalanceWhiteAutoDamping
- quickSpinIntegerNode AasRoiHeight
- quickSpinFloatNode AutoExposureGreyValueUpperLimit
- quickSpinFloatNode AutoExposureTargetGreyValue
- quickSpinFloatNode AutoExposureGainLowerLimit
- quickSpinFloatNode AutoExposureGreyValueLowerLimit
- quickSpinEnumerationNode AutoExposureMeteringMode
- quickSpinFloatNode AutoExposureExposureTimeUpperLimit
- quickSpinFloatNode AutoExposureGainUpperLimit

- [quickSpinFloatNode AutoExposureControlLoopDamping](#)
- [quickSpinFloatNode AutoExposureEVCompensation](#)
- [quickSpinFloatNode AutoExposureExposureTimeLowerLimit](#)
- [quickSpinEnumerationNode BalanceWhiteAutoProfile](#)
- [quickSpinEnumerationNode AutoAlgorithmSelector](#)
- [quickSpinEnumerationNode AutoExposureTargetGreyValueAuto](#)
- [quickSpinBooleanNode AasRoiEnable](#)
- [quickSpinEnumerationNode AutoExposureLightingMode](#)
- [quickSpinIntegerNode AasRoiWidth](#)
- [quickSpinFloatNode BalanceWhiteAutoUpperLimit](#)
- [quickSpinIntegerNode LinkErrorCount](#)
- [quickSpinBooleanNode GevCurrentIPConfigurationDHCP](#)
- [quickSpinIntegerNode GevInterfaceSelector](#)
- [quickSpinIntegerNode GevSCPD](#)
- [quickSpinIntegerNode GevTimestampTickFrequency](#)
- [quickSpinIntegerNode GevSCPSPacketSize](#)
- [quickSpinIntegerNode GevCurrentDefaultGateway](#)
- [quickSpinBooleanNode GevSCCFGUnconditionalStreaming](#)
- [quickSpinIntegerNode GevMCTT](#)
- [quickSpinBooleanNode GevSCPSDoNotFragment](#)
- [quickSpinIntegerNode GevCurrentSubnetMask](#)
- [quickSpinIntegerNode GevStreamChannelSelector](#)
- [quickSpinIntegerNode GevCurrentIPAddress](#)
- [quickSpinIntegerNode GevMCSP](#)
- [quickSpinIntegerNode GevGVCPPendingTimeout](#)
- [quickSpinEnumerationNode GevIEEE1588Status](#)
- [quickSpinStringNode GevFirstURL](#)
- [quickSpinIntegerNode GevMACAddress](#)
- [quickSpinIntegerNode GevPersistentSubnetMask](#)
- [quickSpinIntegerNode GevMCPHostPort](#)
- [quickSpinIntegerNode GevSCPHostPort](#)
- [quickSpinBooleanNode GevGVCPPendingAck](#)
- [quickSpinIntegerNode GevSCPInterfaceIndex](#)
- [quickSpinBooleanNode GevSupportedOption](#)
- [quickSpinEnumerationNode GevIEEE1588Mode](#)
- [quickSpinBooleanNode GevCurrentIPConfigurationLLA](#)
- [quickSpinIntegerNode GevSCSP](#)
- [quickSpinBooleanNode GevIEEE1588](#)
- [quickSpinBooleanNode GevSCCFGExtendedChunkData](#)
- [quickSpinIntegerNode GevPersistentIPAddress](#)
- [quickSpinBooleanNode GevCurrentIPConfigurationPersistentIP](#)
- [quickSpinEnumerationNode GevIEEE1588ClockAccuracy](#)
- [quickSpinIntegerNode GevHeartbeatTimeout](#)
- [quickSpinIntegerNode GevPersistentDefaultGateway](#)
- [quickSpinEnumerationNode GevCCP](#)
- [quickSpinIntegerNode GevMCDA](#)
- [quickSpinIntegerNode GevSCDA](#)
- [quickSpinIntegerNode GevSCPDirection](#)
- [quickSpinBooleanNode GevSCPSFireTestPacket](#)
- [quickSpinStringNode GevSecondURL](#)
- [quickSpinEnumerationNode GevSupportedOptionSelector](#)
- [quickSpinBooleanNode GevGVCPHeartbeatDisable](#)
- [quickSpinIntegerNode GevMCRC](#)
- [quickSpinBooleanNode GevSCPSBigEndian](#)
- [quickSpinIntegerNode GevNumberOfInterfaces](#)

- quickSpinIntegerNode TLParamsLocked
- quickSpinIntegerNode PayloadSize
- quickSpinIntegerNode PacketResendRequestCount
- quickSpinBooleanNode SharpeningEnable
- quickSpinEnumerationNode BlackLevelSelector
- quickSpinBooleanNode GammaEnable
- quickSpinBooleanNode SharpeningAuto
- quickSpinBooleanNode BlackLevelClampingEnable
- quickSpinFloatNode BalanceRatio
- quickSpinEnumerationNode BalanceWhiteAuto
- quickSpinFloatNode SharpeningThreshold
- quickSpinEnumerationNode GainAuto
- quickSpinFloatNode Sharpening
- quickSpinFloatNode Gain
- quickSpinEnumerationNode BalanceRatioSelector
- quickSpinEnumerationNode GainSelector
- quickSpinFloatNode BlackLevel
- quickSpinIntegerNode BlackLevelRaw
- quickSpinFloatNode Gamma
- quickSpinIntegerNode DefectTableIndex
- quickSpinCommandNode DefectTableFactoryRestore
- quickSpinIntegerNode DefectTableCoordinateY
- quickSpinCommandNode DefectTableSave
- quickSpinEnumerationNode DefectCorrectionMode
- quickSpinIntegerNode DefectTableCoordinateX
- quickSpinIntegerNode DefectTablePixelCount
- quickSpinBooleanNode DefectCorrectStaticEnable
- quickSpinCommandNode DefectTableApply
- quickSpinBooleanNode UserSetFeatureEnable
- quickSpinCommandNode UserSetSave
- quickSpinEnumerationNode UserSetSelector
- quickSpinCommandNode UserSetLoad
- quickSpinEnumerationNode UserSetDefault
- quickSpinEnumerationNode SerialPortBaudRate
- quickSpinIntegerNode SerialPortDataBits
- quickSpinEnumerationNode SerialPortParity
- quickSpinIntegerNode SerialTransmitQueueMaxCharacterCount
- quickSpinIntegerNode SerialReceiveQueueCurrentCharacterCount
- quickSpinEnumerationNode SerialPortSelector
- quickSpinEnumerationNode SerialPortStopBits
- quickSpinCommandNode SerialReceiveQueueClear
- quickSpinIntegerNode SerialReceiveFramingErrorCount
- quickSpinIntegerNode SerialTransmitQueueCurrentCharacterCount
- quickSpinIntegerNode SerialReceiveParityErrorCount
- quickSpinEnumerationNode SerialPortSource
- quickSpinIntegerNode SerialReceiveQueueMaxCharacterCount
- quickSpinIntegerNode SequencerSetStart
- quickSpinEnumerationNode SequencerMode
- quickSpinEnumerationNode SequencerConfigurationValid
- quickSpinEnumerationNode SequencerSetValid
- quickSpinIntegerNode SequencerSetSelector
- quickSpinEnumerationNode SequencerTriggerActivation
- quickSpinEnumerationNode SequencerConfigurationMode
- quickSpinCommandNode SequencerSetSave
- quickSpinEnumerationNode SequencerTriggerSource

- [quickSpinIntegerNode SequencerSetActive](#)
- [quickSpinIntegerNode SequencerSetNext](#)
- [quickSpinCommandNode SequencerSetLoad](#)
- [quickSpinIntegerNode SequencerPathSelector](#)
- [quickSpinBooleanNode SequencerFeatureEnable](#)
- [quickSpinIntegerNode TransferBlockCount](#)
- [quickSpinCommandNode TransferStart](#)
- [quickSpinIntegerNode TransferQueueMaxBlockCount](#)
- [quickSpinIntegerNode TransferQueueCurrentBlockCount](#)
- [quickSpinEnumerationNode TransferQueueMode](#)
- [quickSpinEnumerationNode TransferOperationMode](#)
- [quickSpinCommandNode TransferStop](#)
- [quickSpinIntegerNode TransferQueueOverflowCount](#)
- [quickSpinEnumerationNode TransferControlMode](#)
- [quickSpinFloatNode ChunkBlackLevel](#)
- [quickSpinIntegerNode ChunkFrameID](#)
- [quickSpinStringNode ChunkSerialData](#)
- [quickSpinFloatNode ChunkExposureTime](#)
- [quickSpinBooleanNode ChunkSerialReceiveOverflow](#)
- [quickSpinIntegerNode ChunkTimestamp](#)
- [quickSpinBooleanNode ChunkModeActive](#)
- [quickSpinIntegerNode ChunkExposureEndLineStatusAll](#)
- [quickSpinEnumerationNode ChunkGainSelector](#)
- [quickSpinEnumerationNode ChunkSelector](#)
- [quickSpinEnumerationNode ChunkBlackLevelSelector](#)
- [quickSpinIntegerNode ChunkWidth](#)
- [quickSpinIntegerNode ChunkImage](#)
- [quickSpinIntegerNode ChunkHeight](#)
- [quickSpinEnumerationNode ChunkPixelFormat](#)
- [quickSpinFloatNode ChunkGain](#)
- [quickSpinIntegerNode ChunkSequencerSetActive](#)
- [quickSpinIntegerNode ChunkCRC](#)
- [quickSpinIntegerNode ChunkOffsetX](#)
- [quickSpinIntegerNode ChunkOffsetY](#)
- [quickSpinBooleanNode ChunkEnable](#)
- [quickSpinIntegerNode ChunkSerialDataLength](#)
- [quickSpinIntegerNode FileAccessOffset](#)
- [quickSpinIntegerNode FileAccessLength](#)
- [quickSpinEnumerationNode FileOperationStatus](#)
- [quickSpinCommandNode FileOperationExecute](#)
- [quickSpinEnumerationNode FileOpenMode](#)
- [quickSpinIntegerNode FileOperationResult](#)
- [quickSpinEnumerationNode FileOperationSelector](#)
- [quickSpinEnumerationNode FileSelector](#)
- [quickSpinIntegerNode FileSize](#)
- [quickSpinEnumerationNode BinningSelector](#)
- [quickSpinIntegerNode PixelDynamicRangeMin](#)
- [quickSpinIntegerNode PixelDynamicRangeMax](#)
- [quickSpinIntegerNode OffsetY](#)
- [quickSpinIntegerNode BinningHorizontal](#)
- [quickSpinIntegerNode Width](#)
- [quickSpinEnumerationNode TestPatternGeneratorSelector](#)
- [quickSpinFloatNode CompressionRatio](#)
- [quickSpinBooleanNode ReverseX](#)
- [quickSpinBooleanNode ReverseY](#)



- [quickSpinEnumerationNode TestPattern](#)
- [quickSpinEnumerationNode PixelColorFilter](#)
- [quickSpinIntegerNode WidthMax](#)
- [quickSpinEnumerationNode AdcBitDepth](#)
- [quickSpinIntegerNode BinningVertical](#)
- [quickSpinEnumerationNode DecimationHorizontalMode](#)
- [quickSpinEnumerationNode BinningVerticalMode](#)
- [quickSpinIntegerNode OffsetX](#)
- [quickSpinIntegerNode HeightMax](#)
- [quickSpinIntegerNode DecimationHorizontal](#)
- [quickSpinEnumerationNode PixelSize](#)
- [quickSpinIntegerNode SensorHeight](#)
- [quickSpinEnumerationNode DecimationSelector](#)
- [quickSpinBooleanNode IspEnable](#)
- [quickSpinBooleanNode AdaptiveCompressionEnable](#)
- [quickSpinEnumerationNode ImageCompressionMode](#)
- [quickSpinIntegerNode DecimationVertical](#)
- [quickSpinIntegerNode Height](#)
- [quickSpinEnumerationNode BinningHorizontalMode](#)
- [quickSpinEnumerationNode PixelFormat](#)
- [quickSpinIntegerNode SensorWidth](#)
- [quickSpinEnumerationNode DecimationVerticalMode](#)
- [quickSpinCommandNode TestEventGenerate](#)
- [quickSpinCommandNode TriggerEventTest](#)
- [quickSpinIntegerNode GuiXmlManifestAddress](#)
- [quickSpinIntegerNode Test0001](#)
- [quickSpinBooleanNode V3\\_3Enable](#)
- [quickSpinEnumerationNode LineMode](#)
- [quickSpinEnumerationNode LineSource](#)
- [quickSpinEnumerationNode LineInputFilterSelector](#)
- [quickSpinBooleanNode UserOutputValue](#)
- [quickSpinIntegerNode UserOutputValueAll](#)
- [quickSpinEnumerationNode UserOutputSelector](#)
- [quickSpinBooleanNode LineStatus](#)
- [quickSpinEnumerationNode LineFormat](#)
- [quickSpinIntegerNode LineStatusAll](#)
- [quickSpinEnumerationNode LineSelector](#)
- [quickSpinEnumerationNode ExposureActiveMode](#)
- [quickSpinBooleanNode LineInverter](#)
- [quickSpinFloatNode LineFilterWidth](#)
- [quickSpinEnumerationNode CounterTriggerActivation](#)
- [quickSpinIntegerNode CounterValue](#)
- [quickSpinEnumerationNode CounterSelector](#)
- [quickSpinIntegerNode CounterValueAtReset](#)
- [quickSpinEnumerationNode CounterStatus](#)
- [quickSpinEnumerationNode CounterTriggerSource](#)
- [quickSpinIntegerNode CounterDelay](#)
- [quickSpinEnumerationNode CounterResetSource](#)
- [quickSpinEnumerationNode CounterEventSource](#)
- [quickSpinEnumerationNode CounterEventActivation](#)
- [quickSpinIntegerNode CounterDuration](#)
- [quickSpinEnumerationNode CounterResetActivation](#)
- [quickSpinEnumerationNode DeviceType](#)
- [quickSpinStringNode DeviceFamilyName](#)
- [quickSpinIntegerNode DeviceSFNCVersionMajor](#)

- [quickSpinIntegerNode DeviceSFNCVersionMinor](#)
- [quickSpinIntegerNode DeviceSFNCVersionSubMinor](#)
- [quickSpinIntegerNode DeviceManifestEntrySelector](#)
- [quickSpinIntegerNode DeviceManifestXMLMajorVersion](#)
- [quickSpinIntegerNode DeviceManifestXMLMinorVersion](#)
- [quickSpinIntegerNode DeviceManifestXMLSubMinorVersion](#)
- [quickSpinIntegerNode DeviceManifestSchemaMajorVersion](#)
- [quickSpinIntegerNode DeviceManifestSchemaMinorVersion](#)
- [quickSpinStringNode DeviceManifestPrimaryURL](#)
- [quickSpinStringNode DeviceManifestSecondaryURL](#)
- [quickSpinIntegerNode DeviceTLVersionSubMinor](#)
- [quickSpinIntegerNode DeviceGenCPVersionMajor](#)
- [quickSpinIntegerNode DeviceGenCPVersionMinor](#)
- [quickSpinIntegerNode DeviceConnectionSelector](#)
- [quickSpinIntegerNode DeviceConnectionSpeed](#)
- [quickSpinEnumerationNode DeviceConnectionStatus](#)
- [quickSpinIntegerNode DeviceLinkSelector](#)
- [quickSpinEnumerationNode DeviceLinkThroughputLimitMode](#)
- [quickSpinIntegerNode DeviceLinkConnectionCount](#)
- [quickSpinEnumerationNode DeviceLinkHeartbeatMode](#)
- [quickSpinFloatNode DeviceLinkHeartbeatTimeout](#)
- [quickSpinFloatNode DeviceLinkCommandTimeout](#)
- [quickSpinIntegerNode DeviceStreamChannelSelector](#)
- [quickSpinEnumerationNode DeviceStreamChannelType](#)
- [quickSpinIntegerNode DeviceStreamChannelLink](#)
- [quickSpinEnumerationNode DeviceStreamChannelEndianness](#)
- [quickSpinIntegerNode DeviceStreamChannelPacketSize](#)
- [quickSpinCommandNode DeviceFeaturePersistenceStart](#)
- [quickSpinCommandNode DeviceFeaturePersistenceEnd](#)
- [quickSpinCommandNode DeviceRegistersStreamingStart](#)
- [quickSpinCommandNode DeviceRegistersStreamingEnd](#)
- [quickSpinCommandNode DeviceRegistersCheck](#)
- [quickSpinBooleanNode DeviceRegistersValid](#)
- [quickSpinEnumerationNode DeviceClockSelector](#)
- [quickSpinFloatNode DeviceClockFrequency](#)
- [quickSpinEnumerationNode DeviceSerialPortSelector](#)
- [quickSpinEnumerationNode DeviceSerialPortBaudRate](#)
- [quickSpinIntegerNode Timestamp](#)
- [quickSpinEnumerationNode SensorTaps](#)
- [quickSpinEnumerationNode SensorDigitizationTaps](#)
- [quickSpinEnumerationNode RegionSelector](#)
- [quickSpinEnumerationNode RegionMode](#)
- [quickSpinEnumerationNode RegionDestination](#)
- [quickSpinEnumerationNode ImageComponentSelector](#)
- [quickSpinBooleanNode ImageComponentEnable](#)
- [quickSpinIntegerNode LinePitch](#)
- [quickSpinEnumerationNode PixelFormatInfoSelector](#)
- [quickSpinIntegerNode PixelFormatInfoID](#)
- [quickSpinEnumerationNode Deinterlacing](#)
- [quickSpinEnumerationNode ImageCompressionRateOption](#)
- [quickSpinIntegerNode ImageCompressionQuality](#)
- [quickSpinFloatNode ImageCompressionBitrate](#)
- [quickSpinEnumerationNode ImageCompressionJPEGFormatOption](#)
- [quickSpinCommandNode AcquisitionAbort](#)
- [quickSpinCommandNode AcquisitionArm](#)

- [quickSpinEnumerationNode AcquisitionStatusSelector](#)
- [quickSpinBooleanNode AcquisitionStatus](#)
- [quickSpinIntegerNode TriggerDivider](#)
- [quickSpinIntegerNode TriggerMultiplier](#)
- [quickSpinEnumerationNode ExposureTimeMode](#)
- [quickSpinEnumerationNode ExposureTimeSelector](#)
- [quickSpinEnumerationNode GainAutoBalance](#)
- [quickSpinEnumerationNode BlackLevelAuto](#)
- [quickSpinEnumerationNode BlackLevelAutoBalance](#)
- [quickSpinEnumerationNode WhiteClipSelector](#)
- [quickSpinFloatNode WhiteClip](#)
- [quickSpinRegisterNode LUTValueAll](#)
- [quickSpinIntegerNode UserOutputValueAllMask](#)
- [quickSpinCommandNode CounterReset](#)
- [quickSpinEnumerationNode TimerSelector](#)
- [quickSpinFloatNode TimerDuration](#)
- [quickSpinFloatNode TimerDelay](#)
- [quickSpinCommandNode TimerReset](#)
- [quickSpinFloatNode TimerValue](#)
- [quickSpinEnumerationNode TimerStatus](#)
- [quickSpinEnumerationNode TimerTriggerSource](#)
- [quickSpinEnumerationNode TimerTriggerActivation](#)
- [quickSpinEnumerationNode EncoderSelector](#)
- [quickSpinEnumerationNode EncoderSourceA](#)
- [quickSpinEnumerationNode EncoderSourceB](#)
- [quickSpinEnumerationNode EncoderMode](#)
- [quickSpinIntegerNode EncoderDivider](#)
- [quickSpinEnumerationNode EncoderOutputMode](#)
- [quickSpinEnumerationNode EncoderStatus](#)
- [quickSpinFloatNode EncoderTimeout](#)
- [quickSpinEnumerationNode EncoderResetSource](#)
- [quickSpinEnumerationNode EncoderResetActivation](#)
- [quickSpinCommandNode EncoderReset](#)
- [quickSpinIntegerNode EncoderValue](#)
- [quickSpinIntegerNode EncoderValueAtReset](#)
- [quickSpinEnumerationNode SoftwareSignalSelector](#)
- [quickSpinCommandNode SoftwareSignalPulse](#)
- [quickSpinEnumerationNode ActionUnconditionalMode](#)
- [quickSpinIntegerNode ActionDeviceKey](#)
- [quickSpinIntegerNode ActionQueueSize](#)
- [quickSpinIntegerNode ActionSelector](#)
- [quickSpinIntegerNode ActionGroupMask](#)
- [quickSpinIntegerNode ActionGroupKey](#)
- [quickSpinIntegerNode EventAcquisitionTrigger](#)
- [quickSpinIntegerNode EventAcquisitionTriggerTimestamp](#)
- [quickSpinIntegerNode EventAcquisitionTriggerFrameID](#)
- [quickSpinIntegerNode EventAcquisitionStart](#)
- [quickSpinIntegerNode EventAcquisitionStartTimestamp](#)
- [quickSpinIntegerNode EventAcquisitionStartFrameID](#)
- [quickSpinIntegerNode EventAcquisitionEnd](#)
- [quickSpinIntegerNode EventAcquisitionEndTimestamp](#)
- [quickSpinIntegerNode EventAcquisitionEndFrameID](#)
- [quickSpinIntegerNode EventAcquisitionTransferStart](#)
- [quickSpinIntegerNode EventAcquisitionTransferStartTimestamp](#)
- [quickSpinIntegerNode EventAcquisitionTransferStartFrameID](#)

- [quickSpinIntegerNode EventAcquisitionTransferEnd](#)
- [quickSpinIntegerNode EventAcquisitionTransferEndTimestamp](#)
- [quickSpinIntegerNode EventAcquisitionTransferEndFrameID](#)
- [quickSpinIntegerNode EventAcquisitionError](#)
- [quickSpinIntegerNode EventAcquisitionErrorTimestamp](#)
- [quickSpinIntegerNode EventAcquisitionErrorFrameID](#)
- [quickSpinIntegerNode EventFrameTrigger](#)
- [quickSpinIntegerNode EventFrameTriggerTimestamp](#)
- [quickSpinIntegerNode EventFrameTriggerFrameID](#)
- [quickSpinIntegerNode EventFrameStart](#)
- [quickSpinIntegerNode EventFrameStartTimestamp](#)
- [quickSpinIntegerNode EventFrameStartFrameID](#)
- [quickSpinIntegerNode EventFrameEnd](#)
- [quickSpinIntegerNode EventFrameEndTimestamp](#)
- [quickSpinIntegerNode EventFrameEndFrameID](#)
- [quickSpinIntegerNode EventFrameBurstStart](#)
- [quickSpinIntegerNode EventFrameBurstStartTimestamp](#)
- [quickSpinIntegerNode EventFrameBurstStartFrameID](#)
- [quickSpinIntegerNode EventFrameBurstEnd](#)
- [quickSpinIntegerNode EventFrameBurstEndTimestamp](#)
- [quickSpinIntegerNode EventFrameBurstEndFrameID](#)
- [quickSpinIntegerNode EventFrameTransferStart](#)
- [quickSpinIntegerNode EventFrameTransferStartTimestamp](#)
- [quickSpinIntegerNode EventFrameTransferStartFrameID](#)
- [quickSpinIntegerNode EventFrameTransferEnd](#)
- [quickSpinIntegerNode EventFrameTransferEndTimestamp](#)
- [quickSpinIntegerNode EventFrameTransferEndFrameID](#)
- [quickSpinIntegerNode EventExposureStart](#)
- [quickSpinIntegerNode EventExposureStartTimestamp](#)
- [quickSpinIntegerNode EventExposureStartFrameID](#)
- [quickSpinIntegerNode EventStream0TransferStart](#)
- [quickSpinIntegerNode EventStream0TransferStartTimestamp](#)
- [quickSpinIntegerNode EventStream0TransferStartFrameID](#)
- [quickSpinIntegerNode EventStream0TransferEnd](#)
- [quickSpinIntegerNode EventStream0TransferEndTimestamp](#)
- [quickSpinIntegerNode EventStream0TransferEndFrameID](#)
- [quickSpinIntegerNode EventStream0TransferPause](#)
- [quickSpinIntegerNode EventStream0TransferPauseTimestamp](#)
- [quickSpinIntegerNode EventStream0TransferPauseFrameID](#)
- [quickSpinIntegerNode EventStream0TransferResume](#)
- [quickSpinIntegerNode EventStream0TransferResumeTimestamp](#)
- [quickSpinIntegerNode EventStream0TransferResumeFrameID](#)
- [quickSpinIntegerNode EventStream0TransferBlockStart](#)
- [quickSpinIntegerNode EventStream0TransferBlockStartTimestamp](#)
- [quickSpinIntegerNode EventStream0TransferBlockStartFrameID](#)
- [quickSpinIntegerNode EventStream0TransferBlockEnd](#)
- [quickSpinIntegerNode EventStream0TransferBlockEndTimestamp](#)
- [quickSpinIntegerNode EventStream0TransferBlockEndFrameID](#)
- [quickSpinIntegerNode EventStream0TransferBlockTrigger](#)
- [quickSpinIntegerNode EventStream0TransferBlockTriggerTimestamp](#)
- [quickSpinIntegerNode EventStream0TransferBlockTriggerFrameID](#)
- [quickSpinIntegerNode EventStream0TransferBurstStart](#)
- [quickSpinIntegerNode EventStream0TransferBurstStartTimestamp](#)
- [quickSpinIntegerNode EventStream0TransferBurstStartFrameID](#)
- [quickSpinIntegerNode EventStream0TransferBurstEnd](#)

- [quickSpinIntegerNode EventStream0TransferBurstEndTimestamp](#)
- [quickSpinIntegerNode EventStream0TransferBurstEndFrameID](#)
- [quickSpinIntegerNode EventStream0TransferOverflow](#)
- [quickSpinIntegerNode EventStream0TransferOverflowTimestamp](#)
- [quickSpinIntegerNode EventStream0TransferOverflowFrameID](#)
- [quickSpinIntegerNode EventSequencerSetChange](#)
- [quickSpinIntegerNode EventSequencerSetChangeTimestamp](#)
- [quickSpinIntegerNode EventSequencerSetChangeFrameID](#)
- [quickSpinIntegerNode EventCounter0Start](#)
- [quickSpinIntegerNode EventCounter0StartTimestamp](#)
- [quickSpinIntegerNode EventCounter0StartFrameID](#)
- [quickSpinIntegerNode EventCounter1Start](#)
- [quickSpinIntegerNode EventCounter1StartTimestamp](#)
- [quickSpinIntegerNode EventCounter1StartFrameID](#)
- [quickSpinIntegerNode EventCounter0End](#)
- [quickSpinIntegerNode EventCounter0EndTimestamp](#)
- [quickSpinIntegerNode EventCounter0EndFrameID](#)
- [quickSpinIntegerNode EventCounter1End](#)
- [quickSpinIntegerNode EventCounter1EndTimestamp](#)
- [quickSpinIntegerNode EventCounter1EndFrameID](#)
- [quickSpinIntegerNode EventTimer0Start](#)
- [quickSpinIntegerNode EventTimer0StartTimestamp](#)
- [quickSpinIntegerNode EventTimer0StartFrameID](#)
- [quickSpinIntegerNode EventTimer1Start](#)
- [quickSpinIntegerNode EventTimer1StartTimestamp](#)
- [quickSpinIntegerNode EventTimer1StartFrameID](#)
- [quickSpinIntegerNode EventTimer0End](#)
- [quickSpinIntegerNode EventTimer0EndTimestamp](#)
- [quickSpinIntegerNode EventTimer0EndFrameID](#)
- [quickSpinIntegerNode EventTimer1End](#)
- [quickSpinIntegerNode EventTimer1EndTimestamp](#)
- [quickSpinIntegerNode EventTimer1EndFrameID](#)
- [quickSpinIntegerNode EventEncoder0Stopped](#)
- [quickSpinIntegerNode EventEncoder0StoppedTimestamp](#)
- [quickSpinIntegerNode EventEncoder0StoppedFrameID](#)
- [quickSpinIntegerNode EventEncoder1Stopped](#)
- [quickSpinIntegerNode EventEncoder1StoppedTimestamp](#)
- [quickSpinIntegerNode EventEncoder1StoppedFrameID](#)
- [quickSpinIntegerNode EventEncoder0Restarted](#)
- [quickSpinIntegerNode EventEncoder0RestartedTimestamp](#)
- [quickSpinIntegerNode EventEncoder0RestartedFrameID](#)
- [quickSpinIntegerNode EventEncoder1Restarted](#)
- [quickSpinIntegerNode EventEncoder1RestartedTimestamp](#)
- [quickSpinIntegerNode EventEncoder1RestartedFrameID](#)
- [quickSpinIntegerNode EventLine0RisingEdge](#)
- [quickSpinIntegerNode EventLine0RisingEdgeTimestamp](#)
- [quickSpinIntegerNode EventLine0RisingEdgeFrameID](#)
- [quickSpinIntegerNode EventLine1RisingEdge](#)
- [quickSpinIntegerNode EventLine1RisingEdgeTimestamp](#)
- [quickSpinIntegerNode EventLine1RisingEdgeFrameID](#)
- [quickSpinIntegerNode EventLine0FallingEdge](#)
- [quickSpinIntegerNode EventLine0FallingEdgeTimestamp](#)
- [quickSpinIntegerNode EventLine0FallingEdgeFrameID](#)
- [quickSpinIntegerNode EventLine1FallingEdge](#)
- [quickSpinIntegerNode EventLine1FallingEdgeTimestamp](#)

- [quickSpinIntegerNode EventLine1FallingEdgeFrameID](#)
- [quickSpinIntegerNode EventLine0AnyEdge](#)
- [quickSpinIntegerNode EventLine0AnyEdgeTimestamp](#)
- [quickSpinIntegerNode EventLine0AnyEdgeFrameID](#)
- [quickSpinIntegerNode EventLine1AnyEdge](#)
- [quickSpinIntegerNode EventLine1AnyEdgeTimestamp](#)
- [quickSpinIntegerNode EventLine1AnyEdgeFrameID](#)
- [quickSpinIntegerNode EventLinkTrigger0](#)
- [quickSpinIntegerNode EventLinkTrigger0Timestamp](#)
- [quickSpinIntegerNode EventLinkTrigger0FrameID](#)
- [quickSpinIntegerNode EventLinkTrigger1](#)
- [quickSpinIntegerNode EventLinkTrigger1Timestamp](#)
- [quickSpinIntegerNode EventLinkTrigger1FrameID](#)
- [quickSpinIntegerNode EventActionLate](#)
- [quickSpinIntegerNode EventActionLateTimestamp](#)
- [quickSpinIntegerNode EventActionLateFrameID](#)
- [quickSpinIntegerNode EventLinkSpeedChange](#)
- [quickSpinIntegerNode EventLinkSpeedChangeTimestamp](#)
- [quickSpinIntegerNode EventLinkSpeedChangeFrameID](#)
- [quickSpinRegisterNode FileAccessBuffer](#)
- [quickSpinIntegerNode SourceCount](#)
- [quickSpinEnumerationNode SourceSelector](#)
- [quickSpinEnumerationNode TransferSelector](#)
- [quickSpinIntegerNode TransferBurstCount](#)
- [quickSpinCommandNode TransferAbort](#)
- [quickSpinCommandNode TransferPause](#)
- [quickSpinCommandNode TransferResume](#)
- [quickSpinEnumerationNode TransferTriggerSelector](#)
- [quickSpinEnumerationNode TransferTriggerMode](#)
- [quickSpinEnumerationNode TransferTriggerSource](#)
- [quickSpinEnumerationNode TransferTriggerActivation](#)
- [quickSpinEnumerationNode TransferStatusSelector](#)
- [quickSpinBooleanNode TransferStatus](#)
- [quickSpinEnumerationNode TransferComponentSelector](#)
- [quickSpinIntegerNode TransferStreamChannel](#)
- [quickSpinEnumerationNode Scan3dDistanceUnit](#)
- [quickSpinEnumerationNode Scan3dCoordinateSystem](#)
- [quickSpinEnumerationNode Scan3dOutputMode](#)
- [quickSpinEnumerationNode Scan3dCoordinateSystemReference](#)
- [quickSpinEnumerationNode Scan3dCoordinateSelector](#)
- [quickSpinFloatNode Scan3dCoordinateScale](#)
- [quickSpinFloatNode Scan3dCoordinateOffset](#)
- [quickSpinBooleanNode Scan3dInvalidDataFlag](#)
- [quickSpinFloatNode Scan3dInvalidDataValue](#)
- [quickSpinFloatNode Scan3dAxisMin](#)
- [quickSpinFloatNode Scan3dAxisMax](#)
- [quickSpinEnumerationNode Scan3dCoordinateTransformSelector](#)
- [quickSpinFloatNode Scan3dTransformValue](#)
- [quickSpinEnumerationNode Scan3dCoordinateReferenceSelector](#)
- [quickSpinFloatNode Scan3dCoordinateReferenceValue](#)
- [quickSpinIntegerNode ChunkPartSelector](#)
- [quickSpinEnumerationNode ChunkImageComponent](#)
- [quickSpinIntegerNode ChunkPixelDynamicRangeMin](#)
- [quickSpinIntegerNode ChunkPixelDynamicRangeMax](#)
- [quickSpinIntegerNode ChunkTimestampLatchValue](#)

- [quickSpinIntegerNode ChunkLineStatusAll](#)
- [quickSpinEnumerationNode ChunkCounterSelector](#)
- [quickSpinIntegerNode ChunkCounterValue](#)
- [quickSpinEnumerationNode ChunkTimerSelector](#)
- [quickSpinFloatNode ChunkTimerValue](#)
- [quickSpinEnumerationNode ChunkEncoderSelector](#)
- [quickSpinIntegerNode ChunkScanLineSelector](#)
- [quickSpinIntegerNode ChunkEncoderValue](#)
- [quickSpinEnumerationNode ChunkEncoderStatus](#)
- [quickSpinEnumerationNode ChunkExposureTimeSelector](#)
- [quickSpinIntegerNode ChunkLinePitch](#)
- [quickSpinEnumerationNode ChunkSourceID](#)
- [quickSpinEnumerationNode ChunkRegionID](#)
- [quickSpinIntegerNode ChunkTransferBlockID](#)
- [quickSpinEnumerationNode ChunkTransferStreamID](#)
- [quickSpinIntegerNode ChunkTransferQueueCurrentBlockCount](#)
- [quickSpinIntegerNode ChunkStreamChannelID](#)
- [quickSpinEnumerationNode ChunkScan3dDistanceUnit](#)
- [quickSpinEnumerationNode ChunkScan3dOutputMode](#)
- [quickSpinEnumerationNode ChunkScan3dCoordinateSystem](#)
- [quickSpinEnumerationNode ChunkScan3dCoordinateSystemReference](#)
- [quickSpinEnumerationNode ChunkScan3dCoordinateSelector](#)
- [quickSpinFloatNode ChunkScan3dCoordinateScale](#)
- [quickSpinFloatNode ChunkScan3dCoordinateOffset](#)
- [quickSpinBooleanNode ChunkScan3dInvalidDataFlag](#)
- [quickSpinFloatNode ChunkScan3dInvalidDataValue](#)
- [quickSpinFloatNode ChunkScan3dAxisMin](#)
- [quickSpinFloatNode ChunkScan3dAxisMax](#)
- [quickSpinEnumerationNode ChunkScan3dCoordinateTransformSelector](#)
- [quickSpinFloatNode ChunkScan3dTransformValue](#)
- [quickSpinEnumerationNode ChunkScan3dCoordinateReferenceSelector](#)
- [quickSpinFloatNode ChunkScan3dCoordinateReferenceValue](#)
- [quickSpinIntegerNode TestPendingAck](#)
- [quickSpinEnumerationNode DeviceTapGeometry](#)
- [quickSpinEnumerationNode GevPhysicalLinkConfiguration](#)
- [quickSpinEnumerationNode GevCurrentPhysicalLinkConfiguration](#)
- [quickSpinIntegerNode GevActiveLinkCount](#)
- [quickSpinBooleanNode GevPAUSEFrameReception](#)
- [quickSpinBooleanNode GevPAUSEFrameTransmission](#)
- [quickSpinEnumerationNode GevIPConfigurationStatus](#)
- [quickSpinIntegerNode GevDiscoveryAckDelay](#)
- [quickSpinEnumerationNode GevGVCPExtendedStatusCodesSelector](#)
- [quickSpinBooleanNode GevGVCPExtendedStatusCodes](#)
- [quickSpinIntegerNode GevPrimaryApplicationSwitchoverKey](#)
- [quickSpinEnumerationNode GevGVSPExtendedIDMode](#)
- [quickSpinIntegerNode GevPrimaryApplicationSocket](#)
- [quickSpinIntegerNode GevPrimaryApplicationIPAddress](#)
- [quickSpinBooleanNode GevSCCFGPacketResendDestination](#)
- [quickSpinBooleanNode GevSCCFGAllInTransmission](#)
- [quickSpinIntegerNode GevSCZoneCount](#)
- [quickSpinIntegerNode GevSCZoneDirectionAll](#)
- [quickSpinBooleanNode GevSCZoneConfigurationLock](#)
- [quickSpinIntegerNode aPAUSEMACCtrlFramesTransmitted](#)
- [quickSpinIntegerNode aPAUSEMACCtrlFramesReceived](#)
- [quickSpinEnumerationNode CIconfiguration](#)

- [quickSpinEnumerationNode CTimeSlotsCount](#)
- [quickSpinEnumerationNode CxpLinkConfigurationStatus](#)
- [quickSpinEnumerationNode CxpLinkConfigurationPreferred](#)
- [quickSpinEnumerationNode CxpLinkConfiguration](#)
- [quickSpinIntegerNode CxpConnectionSelector](#)
- [quickSpinEnumerationNode CxpConnectionTestMode](#)
- [quickSpinIntegerNode CxpConnectionTestErrorCount](#)
- [quickSpinIntegerNode CxpConnectionTestPacketCount](#)
- [quickSpinCommandNode CxpPoCxpAuto](#)
- [quickSpinCommandNode CxpPoCxpTurnOff](#)
- [quickSpinCommandNode CxpPoCxpTripReset](#)
- [quickSpinEnumerationNode CxpPoCxpStatus](#)
- [quickSpinIntegerNode ChunkInferenceFrameId](#)
- [quickSpinIntegerNode ChunkInferenceResult](#)
- [quickSpinFloatNode ChunkInferenceConfidence](#)
- [quickSpinRegisterNode ChunkInferenceBoundingBoxResult](#)

## 5.2.1 Field Documentation

### 5.2.1.1 AasRoiEnable

[quickSpinBooleanNode](#) AasRoiEnable

### 5.2.1.2 AasRoiHeight

[quickSpinIntegerNode](#) AasRoiHeight

### 5.2.1.3 AasRoiOffsetX

[quickSpinIntegerNode](#) AasRoiOffsetX

### 5.2.1.4 AasRoiOffsetY

[quickSpinIntegerNode](#) AasRoiOffsetY



#### 5.2.1.5 AasRoiWidth

`quickSpinIntegerNode` AasRoiWidth

#### 5.2.1.6 AcquisitionAbort

`quickSpinCommandNode` AcquisitionAbort

#### 5.2.1.7 AcquisitionArm

`quickSpinCommandNode` AcquisitionArm

#### 5.2.1.8 AcquisitionBurstFrameCount

`quickSpinIntegerNode` AcquisitionBurstFrameCount

#### 5.2.1.9 AcquisitionFrameCount

`quickSpinIntegerNode` AcquisitionFrameCount

#### 5.2.1.10 AcquisitionFrameRate

`quickSpinFloatNode` AcquisitionFrameRate

#### 5.2.1.11 AcquisitionFrameRateEnable

`quickSpinBooleanNode` AcquisitionFrameRateEnable

#### 5.2.1.12 AcquisitionLineRate

`quickSpinFloatNode` AcquisitionLineRate

#### 5.2.1.13 AcquisitionMode

`quickSpinEnumerationNode` AcquisitionMode

#### 5.2.1.14 AcquisitionResultingFrameRate

`quickSpinFloatNode` AcquisitionResultingFrameRate

#### 5.2.1.15 AcquisitionStart

`quickSpinCommandNode` AcquisitionStart

#### 5.2.1.16 AcquisitionStatus

`quickSpinBooleanNode` AcquisitionStatus

#### 5.2.1.17 AcquisitionStatusSelector

`quickSpinEnumerationNode` AcquisitionStatusSelector

#### 5.2.1.18 AcquisitionStop

`quickSpinCommandNode` AcquisitionStop

#### 5.2.1.19 ActionDeviceKey

`quickSpinIntegerNode` ActionDeviceKey

#### 5.2.1.20 ActionGroupKey

`quickSpinIntegerNode` ActionGroupKey

#### 5.2.1.21 ActionGroupMask

`quickSpinIntegerNode` ActionGroupMask

#### 5.2.1.22 ActionQueueSize

`quickSpinIntegerNode` ActionQueueSize

#### 5.2.1.23 ActionSelector

`quickSpinIntegerNode` ActionSelector

#### 5.2.1.24 ActionUnconditionalMode

`quickSpinEnumerationNode` ActionUnconditionalMode

#### 5.2.1.25 AdaptiveCompressionEnable

`quickSpinBooleanNode` AdaptiveCompressionEnable

#### 5.2.1.26 AdcBitDepth

`quickSpinEnumerationNode` AdcBitDepth

#### 5.2.1.27 aPAUSEMACCtrlFramesReceived

`quickSpinIntegerNode` aPAUSEMACCtrlFramesReceived

#### 5.2.1.28 aPAUSEMACCtrlFramesTransmitted

`quickSpinIntegerNode` aPAUSEMACCtrlFramesTransmitted

#### 5.2.1.29 AutoAlgorithmSelector

`quickSpinEnumerationNode` AutoAlgorithmSelector

#### 5.2.1.30 AutoExposureControlLoopDamping

`quickSpinFloatNode` AutoExposureControlLoopDamping

#### 5.2.1.31 AutoExposureControlPriority

`quickSpinEnumerationNode` AutoExposureControlPriority

#### 5.2.1.32 AutoExposureEVCompensation

`quickSpinFloatNode` AutoExposureEVCompensation

#### 5.2.1.33 AutoExposureExposureTimeLowerLimit

`quickSpinFloatNode` AutoExposureExposureTimeLowerLimit

#### 5.2.1.34 AutoExposureExposureTimeUpperLimit

`quickSpinFloatNode` AutoExposureExposureTimeUpperLimit

#### 5.2.1.35 AutoExposureGainLowerLimit

`quickSpinFloatNode` AutoExposureGainLowerLimit

#### 5.2.1.36 AutoExposureGainUpperLimit

`quickSpinFloatNode` AutoExposureGainUpperLimit

#### 5.2.1.37 AutoExposureGreyValueLowerLimit

`quickSpinFloatNode` AutoExposureGreyValueLowerLimit

#### 5.2.1.38 AutoExposureGreyValueUpperLimit

`quickSpinFloatNode` AutoExposureGreyValueUpperLimit

#### 5.2.1.39 AutoExposureLightingMode

`quickSpinEnumerationNode` AutoExposureLightingMode

#### 5.2.1.40 AutoExposureMeteringMode

`quickSpinEnumerationNode` AutoExposureMeteringMode

#### 5.2.1.41 AutoExposureTargetGreyValue

`quickSpinFloatNode` AutoExposureTargetGreyValue

#### 5.2.1.42 AutoExposureTargetGreyValueAuto

`quickSpinEnumerationNode` AutoExposureTargetGreyValueAuto

#### 5.2.1.43 BalanceRatio

`quickSpinFloatNode` BalanceRatio

#### 5.2.1.44 BalanceRatioSelector

`quickSpinEnumerationNode` BalanceRatioSelector

#### 5.2.1.45 BalanceWhiteAuto

`quickSpinEnumerationNode` BalanceWhiteAuto

#### 5.2.1.46 BalanceWhiteAutoDamping

`quickSpinFloatNode` BalanceWhiteAutoDamping

#### 5.2.1.47 BalanceWhiteAutoLowerLimit

`quickSpinFloatNode` BalanceWhiteAutoLowerLimit

#### 5.2.1.48 BalanceWhiteAutoProfile

`quickSpinEnumerationNode` BalanceWhiteAutoProfile

#### 5.2.1.49 BalanceWhiteAutoUpperLimit

`quickSpinFloatNode` BalanceWhiteAutoUpperLimit

#### 5.2.1.50 BinningHorizontal

`quickSpinIntegerNode` BinningHorizontal

#### 5.2.1.51 BinningHorizontalMode

`quickSpinEnumerationNode` BinningHorizontalMode

#### 5.2.1.52 BinningSelector

`quickSpinEnumerationNode` BinningSelector

### 5.2.1.53 BinningVertical

`quickSpinIntegerNode` BinningVertical

### 5.2.1.54 BinningVerticalMode

`quickSpinEnumerationNode` BinningVerticalMode

### 5.2.1.55 BlackLevel

`quickSpinFloatNode` BlackLevel

### 5.2.1.56 BlackLevelAuto

`quickSpinEnumerationNode` BlackLevelAuto

### 5.2.1.57 BlackLevelAutoBalance

`quickSpinEnumerationNode` BlackLevelAutoBalance

### 5.2.1.58 BlackLevelClampingEnable

`quickSpinBooleanNode` BlackLevelClampingEnable

### 5.2.1.59 BlackLevelRaw

`quickSpinIntegerNode` BlackLevelRaw

### 5.2.1.60 BlackLevelSelector

`quickSpinEnumerationNode` BlackLevelSelector

#### 5.2.1.61 ChunkBlackLevel

`quickSpinFloatNode` ChunkBlackLevel

#### 5.2.1.62 ChunkBlackLevelSelector

`quickSpinEnumerationNode` ChunkBlackLevelSelector

#### 5.2.1.63 ChunkCounterSelector

`quickSpinEnumerationNode` ChunkCounterSelector

#### 5.2.1.64 ChunkCounterValue

`quickSpinIntegerNode` ChunkCounterValue

#### 5.2.1.65 ChunkCRC

`quickSpinIntegerNode` ChunkCRC

#### 5.2.1.66 ChunkEnable

`quickSpinBooleanNode` ChunkEnable

#### 5.2.1.67 ChunkEncoderSelector

`quickSpinEnumerationNode` ChunkEncoderSelector

#### 5.2.1.68 ChunkEncoderStatus

`quickSpinEnumerationNode` ChunkEncoderStatus



#### 5.2.1.69 ChunkEncoderValue

`quickSpinIntegerNode` ChunkEncoderValue

#### 5.2.1.70 ChunkExposureEndLineStatusAll

`quickSpinIntegerNode` ChunkExposureEndLineStatusAll

#### 5.2.1.71 ChunkExposureTime

`quickSpinFloatNode` ChunkExposureTime

#### 5.2.1.72 ChunkExposureTimeSelector

`quickSpinEnumerationNode` ChunkExposureTimeSelector

#### 5.2.1.73 ChunkFrameID

`quickSpinIntegerNode` ChunkFrameID

#### 5.2.1.74 ChunkGain

`quickSpinFloatNode` ChunkGain

#### 5.2.1.75 ChunkGainSelector

`quickSpinEnumerationNode` ChunkGainSelector

#### 5.2.1.76 ChunkHeight

`quickSpinIntegerNode` ChunkHeight

#### 5.2.1.77 ChunkImage

[quickSpinIntegerNode](#) ChunkImage

#### 5.2.1.78 ChunkImageComponent

[quickSpinEnumerationNode](#) ChunkImageComponent

#### 5.2.1.79 ChunkInferenceBoundingBoxResult

[quickSpinRegisterNode](#) ChunkInferenceBoundingBoxResult

#### 5.2.1.80 ChunkInferenceConfidence

[quickSpinFloatNode](#) ChunkInferenceConfidence

#### 5.2.1.81 ChunkInferenceFrameId

[quickSpinIntegerNode](#) ChunkInferenceFrameId

#### 5.2.1.82 ChunkInferenceResult

[quickSpinIntegerNode](#) ChunkInferenceResult

#### 5.2.1.83 ChunkLinePitch

[quickSpinIntegerNode](#) ChunkLinePitch

#### 5.2.1.84 ChunkLineStatusAll

[quickSpinIntegerNode](#) ChunkLineStatusAll

#### 5.2.1.85 ChunkModeActive

[quickSpinBooleanNode](#) ChunkModeActive

#### 5.2.1.86 ChunkOffsetX

[quickSpinIntegerNode](#) ChunkOffsetX

#### 5.2.1.87 ChunkOffsetY

[quickSpinIntegerNode](#) ChunkOffsetY

#### 5.2.1.88 ChunkPartSelector

[quickSpinIntegerNode](#) ChunkPartSelector

#### 5.2.1.89 ChunkPixelDynamicRangeMax

[quickSpinIntegerNode](#) ChunkPixelDynamicRangeMax

#### 5.2.1.90 ChunkPixelDynamicRangeMin

[quickSpinIntegerNode](#) ChunkPixelDynamicRangeMin

#### 5.2.1.91 ChunkPixelFormat

[quickSpinEnumerationNode](#) ChunkPixelFormat

#### 5.2.1.92 ChunkRegionID

[quickSpinEnumerationNode](#) ChunkRegionID

#### 5.2.1.93 ChunkScan3dAxisMax

[quickSpinFloatNode](#) ChunkScan3dAxisMax

#### 5.2.1.94 ChunkScan3dAxisMin

[quickSpinFloatNode](#) ChunkScan3dAxisMin

#### 5.2.1.95 ChunkScan3dCoordinateOffset

[quickSpinFloatNode](#) ChunkScan3dCoordinateOffset

#### 5.2.1.96 ChunkScan3dCoordinateReferenceSelector

[quickSpinEnumerationNode](#) ChunkScan3dCoordinateReferenceSelector

#### 5.2.1.97 ChunkScan3dCoordinateReferenceValue

[quickSpinFloatNode](#) ChunkScan3dCoordinateReferenceValue

#### 5.2.1.98 ChunkScan3dCoordinateScale

[quickSpinFloatNode](#) ChunkScan3dCoordinateScale

#### 5.2.1.99 ChunkScan3dCoordinateSelector

[quickSpinEnumerationNode](#) ChunkScan3dCoordinateSelector

#### 5.2.1.100 ChunkScan3dCoordinateSystem

[quickSpinEnumerationNode](#) ChunkScan3dCoordinateSystem

#### 5.2.1.101 ChunkScan3dCoordinateSystemReference

`quickSpinEnumerationNode` ChunkScan3dCoordinateSystemReference

#### 5.2.1.102 ChunkScan3dCoordinateTransformSelector

`quickSpinEnumerationNode` ChunkScan3dCoordinateTransformSelector

#### 5.2.1.103 ChunkScan3dDistanceUnit

`quickSpinEnumerationNode` ChunkScan3dDistanceUnit

#### 5.2.1.104 ChunkScan3dInvalidDataFlag

`quickSpinBooleanNode` ChunkScan3dInvalidDataFlag

#### 5.2.1.105 ChunkScan3dInvalidDataValue

`quickSpinFloatNode` ChunkScan3dInvalidDataValue

#### 5.2.1.106 ChunkScan3dOutputMode

`quickSpinEnumerationNode` ChunkScan3dOutputMode

#### 5.2.1.107 ChunkScan3dTransformValue

`quickSpinFloatNode` ChunkScan3dTransformValue

#### 5.2.1.108 ChunkScanLineSelector

`quickSpinIntegerNode` ChunkScanLineSelector

**5.2.1.109 ChunkSelector**

`quickSpinEnumerationNode` ChunkSelector

**5.2.1.110 ChunkSequencerSetActive**

`quickSpinIntegerNode` ChunkSequencerSetActive

**5.2.1.111 ChunkSerialData**

`quickSpinStringNode` ChunkSerialData

**5.2.1.112 ChunkSerialDataLength**

`quickSpinIntegerNode` ChunkSerialDataLength

**5.2.1.113 ChunkSerialReceiveOverflow**

`quickSpinBooleanNode` ChunkSerialReceiveOverflow

**5.2.1.114 ChunkSourceID**

`quickSpinEnumerationNode` ChunkSourceID

**5.2.1.115 ChunkStreamChannelID**

`quickSpinIntegerNode` ChunkStreamChannelID

**5.2.1.116 ChunkTimerSelector**

`quickSpinEnumerationNode` ChunkTimerSelector

#### 5.2.1.117 ChunkTimerValue

`quickSpinFloatNode` ChunkTimerValue

#### 5.2.1.118 ChunkTimestamp

`quickSpinIntegerNode` ChunkTimestamp

#### 5.2.1.119 ChunkTimestampLatchValue

`quickSpinIntegerNode` ChunkTimestampLatchValue

#### 5.2.1.120 ChunkTransferBlockID

`quickSpinIntegerNode` ChunkTransferBlockID

#### 5.2.1.121 ChunkTransferQueueCurrentBlockCount

`quickSpinIntegerNode` ChunkTransferQueueCurrentBlockCount

#### 5.2.1.122 ChunkTransferStreamID

`quickSpinEnumerationNode` ChunkTransferStreamID

#### 5.2.1.123 ChunkWidth

`quickSpinIntegerNode` ChunkWidth

#### 5.2.1.124 ClConfiguration

`quickSpinEnumerationNode` ClConfiguration

#### 5.2.1.125 CTimeSlotsCount

`quickSpinEnumerationNode` CTimeSlotsCount

#### 5.2.1.126 ColorTransformationEnable

`quickSpinBooleanNode` ColorTransformationEnable

#### 5.2.1.127 ColorTransformationSelector

`quickSpinEnumerationNode` ColorTransformationSelector

#### 5.2.1.128 ColorTransformationValue

`quickSpinFloatNode` ColorTransformationValue

#### 5.2.1.129 ColorTransformationValueSelector

`quickSpinEnumerationNode` ColorTransformationValueSelector

#### 5.2.1.130 CompressionRatio

`quickSpinFloatNode` CompressionRatio

#### 5.2.1.131 CounterDelay

`quickSpinIntegerNode` CounterDelay

#### 5.2.1.132 CounterDuration

`quickSpinIntegerNode` CounterDuration



#### 5.2.1.133 CounterEventActivation

[quickSpinEnumerationNode](#) CounterEventActivation

#### 5.2.1.134 CounterEventSource

[quickSpinEnumerationNode](#) CounterEventSource

#### 5.2.1.135 CounterReset

[quickSpinCommandNode](#) CounterReset

#### 5.2.1.136 CounterResetActivation

[quickSpinEnumerationNode](#) CounterResetActivation

#### 5.2.1.137 CounterResetSource

[quickSpinEnumerationNode](#) CounterResetSource

#### 5.2.1.138 CounterSelector

[quickSpinEnumerationNode](#) CounterSelector

#### 5.2.1.139 CounterStatus

[quickSpinEnumerationNode](#) CounterStatus

#### 5.2.1.140 CounterTriggerActivation

[quickSpinEnumerationNode](#) CounterTriggerActivation

**5.2.1.141 CounterTriggerSource**

[quickSpinEnumerationNode](#) CounterTriggerSource

**5.2.1.142 CounterValue**

[quickSpinIntegerNode](#) CounterValue

**5.2.1.143 CounterValueAtReset**

[quickSpinIntegerNode](#) CounterValueAtReset

**5.2.1.144 CxpConnectionSelector**

[quickSpinIntegerNode](#) CxpConnectionSelector

**5.2.1.145 CxpConnectionTestErrorCount**

[quickSpinIntegerNode](#) CxpConnectionTestErrorCount

**5.2.1.146 CxpConnectionTestMode**

[quickSpinEnumerationNode](#) CxpConnectionTestMode

**5.2.1.147 CxpConnectionTestPacketCount**

[quickSpinIntegerNode](#) CxpConnectionTestPacketCount

**5.2.1.148 CxpLinkConfiguration**

[quickSpinEnumerationNode](#) CxpLinkConfiguration

#### 5.2.1.149 CxpLinkConfigurationPreferred

[quickSpinEnumerationNode](#) CxpLinkConfigurationPreferred

#### 5.2.1.150 CxpLinkConfigurationStatus

[quickSpinEnumerationNode](#) CxpLinkConfigurationStatus

#### 5.2.1.151 CxpPoCxpAuto

[quickSpinCommandNode](#) CxpPoCxpAuto

#### 5.2.1.152 CxpPoCxpStatus

[quickSpinEnumerationNode](#) CxpPoCxpStatus

#### 5.2.1.153 CxpPoCxpTripReset

[quickSpinCommandNode](#) CxpPoCxpTripReset

#### 5.2.1.154 CxpPoCxpTurnOff

[quickSpinCommandNode](#) CxpPoCxpTurnOff

#### 5.2.1.155 DecimationHorizontal

[quickSpinIntegerNode](#) DecimationHorizontal

#### 5.2.1.156 DecimationHorizontalMode

[quickSpinEnumerationNode](#) DecimationHorizontalMode

**5.2.1.157 DecimationSelector**

`quickSpinEnumerationNode` DecimationSelector

**5.2.1.158 DecimationVertical**

`quickSpinIntegerNode` DecimationVertical

**5.2.1.159 DecimationVerticalMode**

`quickSpinEnumerationNode` DecimationVerticalMode

**5.2.1.160 DefectCorrectionMode**

`quickSpinEnumerationNode` DefectCorrectionMode

**5.2.1.161 DefectCorrectStaticEnable**

`quickSpinBooleanNode` DefectCorrectStaticEnable

**5.2.1.162 DefectTableApply**

`quickSpinCommandNode` DefectTableApply

**5.2.1.163 DefectTableCoordinateX**

`quickSpinIntegerNode` DefectTableCoordinateX

**5.2.1.164 DefectTableCoordinateY**

`quickSpinIntegerNode` DefectTableCoordinateY

#### 5.2.1.165 DefectTableFactoryRestore

`quickSpinCommandNode` DefectTableFactoryRestore

#### 5.2.1.166 DefectTableIndex

`quickSpinIntegerNode` DefectTableIndex

#### 5.2.1.167 DefectTablePixelCount

`quickSpinIntegerNode` DefectTablePixelCount

#### 5.2.1.168 DefectTableSave

`quickSpinCommandNode` DefectTableSave

#### 5.2.1.169 Deinterlacing

`quickSpinEnumerationNode` Deinterlacing

#### 5.2.1.170 DeviceCharacterSet

`quickSpinEnumerationNode` DeviceCharacterSet

#### 5.2.1.171 DeviceClockFrequency

`quickSpinFloatNode` DeviceClockFrequency

#### 5.2.1.172 DeviceClockSelector

`quickSpinEnumerationNode` DeviceClockSelector

**5.2.1.173 DeviceConnectionSelector**

[quickSpinIntegerNode](#) DeviceConnectionSelector

**5.2.1.174 DeviceConnectionSpeed**

[quickSpinIntegerNode](#) DeviceConnectionSpeed

**5.2.1.175 DeviceConnectionStatus**

[quickSpinEnumerationNode](#) DeviceConnectionStatus

**5.2.1.176 DeviceEventChannelCount**

[quickSpinIntegerNode](#) DeviceEventChannelCount

**5.2.1.177 DeviceFamilyName**

[quickSpinStringNode](#) DeviceFamilyName

**5.2.1.178 DeviceFeaturePersistenceEnd**

[quickSpinCommandNode](#) DeviceFeaturePersistenceEnd

**5.2.1.179 DeviceFeaturePersistenceStart**

[quickSpinCommandNode](#) DeviceFeaturePersistenceStart

**5.2.1.180 DeviceFirmwareVersion**

[quickSpinStringNode](#) DeviceFirmwareVersion

**5.2.1.181 DeviceGenCPVersionMajor**

`quickSpinIntegerNode` DeviceGenCPVersionMajor

**5.2.1.182 DeviceGenCPVersionMinor**

`quickSpinIntegerNode` DeviceGenCPVersionMinor

**5.2.1.183 DeviceID**

`quickSpinStringNode` DeviceID

**5.2.1.184 DeviceIndicatorMode**

`quickSpinEnumerationNode` DeviceIndicatorMode

**5.2.1.185 DeviceLinkBandwidthReserve**

`quickSpinFloatNode` DeviceLinkBandwidthReserve

**5.2.1.186 DeviceLinkCommandTimeout**

`quickSpinFloatNode` DeviceLinkCommandTimeout

**5.2.1.187 DeviceLinkConnectionCount**

`quickSpinIntegerNode` DeviceLinkConnectionCount

**5.2.1.188 DeviceLinkCurrentThroughput**

`quickSpinIntegerNode` DeviceLinkCurrentThroughput

#### 5.2.1.189 DeviceLinkHeartbeatMode

`quickSpinEnumerationNode` DeviceLinkHeartbeatMode

#### 5.2.1.190 DeviceLinkHeartbeatTimeout

`quickSpinFloatNode` DeviceLinkHeartbeatTimeout

#### 5.2.1.191 DeviceLinkSelector

`quickSpinIntegerNode` DeviceLinkSelector

#### 5.2.1.192 DeviceLinkSpeed

`quickSpinIntegerNode` DeviceLinkSpeed

#### 5.2.1.193 DeviceLinkThroughputLimit

`quickSpinIntegerNode` DeviceLinkThroughputLimit

#### 5.2.1.194 DeviceLinkThroughputLimitMode

`quickSpinEnumerationNode` DeviceLinkThroughputLimitMode

#### 5.2.1.195 DeviceManifestEntrySelector

`quickSpinIntegerNode` DeviceManifestEntrySelector

#### 5.2.1.196 DeviceManifestPrimaryURL

`quickSpinStringNode` DeviceManifestPrimaryURL



#### 5.2.1.197 DeviceManifestSchemaMajorVersion

`quickSpinIntegerNode` DeviceManifestSchemaMajorVersion

#### 5.2.1.198 DeviceManifestSchemaMinorVersion

`quickSpinIntegerNode` DeviceManifestSchemaMinorVersion

#### 5.2.1.199 DeviceManifestSecondaryURL

`quickSpinStringNode` DeviceManifestSecondaryURL

#### 5.2.1.200 DeviceManifestXMLMajorVersion

`quickSpinIntegerNode` DeviceManifestXMLMajorVersion

#### 5.2.1.201 DeviceManifestXMLMinorVersion

`quickSpinIntegerNode` DeviceManifestXMLMinorVersion

#### 5.2.1.202 DeviceManifestXMLSubMinorVersion

`quickSpinIntegerNode` DeviceManifestXMLSubMinorVersion

#### 5.2.1.203 DeviceManufacturerInfo

`quickSpinStringNode` DeviceManufacturerInfo

#### 5.2.1.204 DeviceMaxThroughput

`quickSpinIntegerNode` DeviceMaxThroughput

**5.2.1.205 DeviceModelName**

`quickSpinStringNode` DeviceModelName

**5.2.1.206 DevicePowerSupplySelector**

`quickSpinEnumerationNode` DevicePowerSupplySelector

**5.2.1.207 DeviceRegistersCheck**

`quickSpinCommandNode` DeviceRegistersCheck

**5.2.1.208 DeviceRegistersEndianness**

`quickSpinEnumerationNode` DeviceRegistersEndianness

**5.2.1.209 DeviceRegistersStreamingEnd**

`quickSpinCommandNode` DeviceRegistersStreamingEnd

**5.2.1.210 DeviceRegistersStreamingStart**

`quickSpinCommandNode` DeviceRegistersStreamingStart

**5.2.1.211 DeviceRegistersValid**

`quickSpinBooleanNode` DeviceRegistersValid

**5.2.1.212 DeviceReset**

`quickSpinCommandNode` DeviceReset

#### 5.2.1.213 DeviceScanType

`quickSpinEnumerationNode` DeviceScanType

#### 5.2.1.214 DeviceSerialNumber

`quickSpinStringNode` DeviceSerialNumber

#### 5.2.1.215 DeviceSerialPortBaudRate

`quickSpinEnumerationNode` DeviceSerialPortBaudRate

#### 5.2.1.216 DeviceSerialPortSelector

`quickSpinEnumerationNode` DeviceSerialPortSelector

#### 5.2.1.217 DeviceSFNCVersionMajor

`quickSpinIntegerNode` DeviceSFNCVersionMajor

#### 5.2.1.218 DeviceSFNCVersionMinor

`quickSpinIntegerNode` DeviceSFNCVersionMinor

#### 5.2.1.219 DeviceSFNCVersionSubMinor

`quickSpinIntegerNode` DeviceSFNCVersionSubMinor

#### 5.2.1.220 DeviceStreamChannelCount

`quickSpinIntegerNode` DeviceStreamChannelCount

**5.2.1.221 DeviceStreamChannelEndianness**

`quickSpinEnumerationNode` DeviceStreamChannelEndianness

**5.2.1.222 DeviceStreamChannelLink**

`quickSpinIntegerNode` DeviceStreamChannelLink

**5.2.1.223 DeviceStreamChannelPacketSize**

`quickSpinIntegerNode` DeviceStreamChannelPacketSize

**5.2.1.224 DeviceStreamChannelSelector**

`quickSpinIntegerNode` DeviceStreamChannelSelector

**5.2.1.225 DeviceStreamChannelType**

`quickSpinEnumerationNode` DeviceStreamChannelType

**5.2.1.226 DeviceTapGeometry**

`quickSpinEnumerationNode` DeviceTapGeometry

**5.2.1.227 DeviceTemperature**

`quickSpinFloatNode` DeviceTemperature

**5.2.1.228 DeviceTemperatureSelector**

`quickSpinEnumerationNode` DeviceTemperatureSelector

#### 5.2.1.229 DeviceTLType

`quickSpinEnumerationNode` DeviceTLType

#### 5.2.1.230 DeviceTLVersionMajor

`quickSpinIntegerNode` DeviceTLVersionMajor

#### 5.2.1.231 DeviceTLVersionMinor

`quickSpinIntegerNode` DeviceTLVersionMinor

#### 5.2.1.232 DeviceTLVersionSubMinor

`quickSpinIntegerNode` DeviceTLVersionSubMinor

#### 5.2.1.233 DeviceType

`quickSpinEnumerationNode` DeviceType

#### 5.2.1.234 DeviceUptime

`quickSpinIntegerNode` DeviceUptime

#### 5.2.1.235 DeviceUserID

`quickSpinStringNode` DeviceUserID

#### 5.2.1.236 DeviceVendorName

`quickSpinStringNode` DeviceVendorName

**5.2.1.237 DeviceVersion**

`quickSpinStringNode` DeviceVersion

**5.2.1.238 EncoderDivider**

`quickSpinIntegerNode` EncoderDivider

**5.2.1.239 EncoderMode**

`quickSpinEnumerationNode` EncoderMode

**5.2.1.240 EncoderOutputMode**

`quickSpinEnumerationNode` EncoderOutputMode

**5.2.1.241 EncoderReset**

`quickSpinCommandNode` EncoderReset

**5.2.1.242 EncoderResetActivation**

`quickSpinEnumerationNode` EncoderResetActivation

**5.2.1.243 EncoderResetSource**

`quickSpinEnumerationNode` EncoderResetSource

**5.2.1.244 EncoderSelector**

`quickSpinEnumerationNode` EncoderSelector

**5.2.1.245 EncoderSourceA**

[quickSpinEnumerationNode](#) EncoderSourceA

**5.2.1.246 EncoderSourceB**

[quickSpinEnumerationNode](#) EncoderSourceB

**5.2.1.247 EncoderStatus**

[quickSpinEnumerationNode](#) EncoderStatus

**5.2.1.248 EncoderTimeout**

[quickSpinFloatNode](#) EncoderTimeout

**5.2.1.249 EncoderValue**

[quickSpinIntegerNode](#) EncoderValue

**5.2.1.250 EncoderValueAtReset**

[quickSpinIntegerNode](#) EncoderValueAtReset

**5.2.1.251 EnumerationCount**

[quickSpinIntegerNode](#) EnumerationCount

**5.2.1.252 EventAcquisitionEnd**

[quickSpinIntegerNode](#) EventAcquisitionEnd

#### 5.2.1.253 EventAcquisitionEndFrameID

`quickSpinIntegerNode` EventAcquisitionEndFrameID

#### 5.2.1.254 EventAcquisitionEndTimestamp

`quickSpinIntegerNode` EventAcquisitionEndTimestamp

#### 5.2.1.255 EventAcquisitionError

`quickSpinIntegerNode` EventAcquisitionError

#### 5.2.1.256 EventAcquisitionErrorFrameID

`quickSpinIntegerNode` EventAcquisitionErrorFrameID

#### 5.2.1.257 EventAcquisitionErrorTimestamp

`quickSpinIntegerNode` EventAcquisitionErrorTimestamp

#### 5.2.1.258 EventAcquisitionStart

`quickSpinIntegerNode` EventAcquisitionStart

#### 5.2.1.259 EventAcquisitionStartFrameID

`quickSpinIntegerNode` EventAcquisitionStartFrameID

#### 5.2.1.260 EventAcquisitionStartTimestamp

`quickSpinIntegerNode` EventAcquisitionStartTimestamp



**5.2.1.261 EventAcquisitionTransferEnd**

`quickSpinIntegerNode` EventAcquisitionTransferEnd

**5.2.1.262 EventAcquisitionTransferEndFrameID**

`quickSpinIntegerNode` EventAcquisitionTransferEndFrameID

**5.2.1.263 EventAcquisitionTransferEndTimestamp**

`quickSpinIntegerNode` EventAcquisitionTransferEndTimestamp

**5.2.1.264 EventAcquisitionTransferStart**

`quickSpinIntegerNode` EventAcquisitionTransferStart

**5.2.1.265 EventAcquisitionTransferStartFrameID**

`quickSpinIntegerNode` EventAcquisitionTransferStartFrameID

**5.2.1.266 EventAcquisitionTransferStartTimestamp**

`quickSpinIntegerNode` EventAcquisitionTransferStartTimestamp

**5.2.1.267 EventAcquisitionTrigger**

`quickSpinIntegerNode` EventAcquisitionTrigger

**5.2.1.268 EventAcquisitionTriggerFrameID**

`quickSpinIntegerNode` EventAcquisitionTriggerFrameID

**5.2.1.269 EventAcquisitionTriggerTimestamp**

`quickSpinIntegerNode` EventAcquisitionTriggerTimestamp

**5.2.1.270 EventActionLate**

`quickSpinIntegerNode` EventActionLate

**5.2.1.271 EventActionLateFrameID**

`quickSpinIntegerNode` EventActionLateFrameID

**5.2.1.272 EventActionLateTimestamp**

`quickSpinIntegerNode` EventActionLateTimestamp

**5.2.1.273 EventCounter0End**

`quickSpinIntegerNode` EventCounter0End

**5.2.1.274 EventCounter0EndFrameID**

`quickSpinIntegerNode` EventCounter0EndFrameID

**5.2.1.275 EventCounter0EndTimestamp**

`quickSpinIntegerNode` EventCounter0EndTimestamp

**5.2.1.276 EventCounter0Start**

`quickSpinIntegerNode` EventCounter0Start

**5.2.1.277 EventCounter0StartFrameID**

`quickSpinIntegerNode` EventCounter0StartFrameID

**5.2.1.278 EventCounter0StartTimestamp**

`quickSpinIntegerNode` EventCounter0StartTimestamp

**5.2.1.279 EventCounter1End**

`quickSpinIntegerNode` EventCounter1End

**5.2.1.280 EventCounter1EndFrameID**

`quickSpinIntegerNode` EventCounter1EndFrameID

**5.2.1.281 EventCounter1EndTimestamp**

`quickSpinIntegerNode` EventCounter1EndTimestamp

**5.2.1.282 EventCounter1Start**

`quickSpinIntegerNode` EventCounter1Start

**5.2.1.283 EventCounter1StartFrameID**

`quickSpinIntegerNode` EventCounter1StartFrameID

**5.2.1.284 EventCounter1StartTimestamp**

`quickSpinIntegerNode` EventCounter1StartTimestamp

**5.2.1.285 EventEncoder0Restarted**

`quickSpinIntegerNode` EventEncoder0Restarted

**5.2.1.286 EventEncoder0RestartedFrameID**

`quickSpinIntegerNode` EventEncoder0RestartedFrameID

**5.2.1.287 EventEncoder0RestartedTimestamp**

`quickSpinIntegerNode` EventEncoder0RestartedTimestamp

**5.2.1.288 EventEncoder0Stopped**

`quickSpinIntegerNode` EventEncoder0Stopped

**5.2.1.289 EventEncoder0StoppedFrameID**

`quickSpinIntegerNode` EventEncoder0StoppedFrameID

**5.2.1.290 EventEncoder0StoppedTimestamp**

`quickSpinIntegerNode` EventEncoder0StoppedTimestamp

**5.2.1.291 EventEncoder1Restarted**

`quickSpinIntegerNode` EventEncoder1Restarted

**5.2.1.292 EventEncoder1RestartedFrameID**

`quickSpinIntegerNode` EventEncoder1RestartedFrameID

**5.2.1.293 EventEncoder1RestartedTimestamp**

`quickSpinIntegerNode` EventEncoder1RestartedTimestamp

**5.2.1.294 EventEncoder1Stopped**

`quickSpinIntegerNode` EventEncoder1Stopped

**5.2.1.295 EventEncoder1StoppedFrameID**

`quickSpinIntegerNode` EventEncoder1StoppedFrameID

**5.2.1.296 EventEncoder1StoppedTimestamp**

`quickSpinIntegerNode` EventEncoder1StoppedTimestamp

**5.2.1.297 EventError**

`quickSpinIntegerNode` EventError

**5.2.1.298 EventErrorCode**

`quickSpinIntegerNode` EventErrorCode

**5.2.1.299 EventErrorFrameID**

`quickSpinIntegerNode` EventErrorFrameID

**5.2.1.300 EventErrorTimestamp**

`quickSpinIntegerNode` EventErrorTimestamp

#### 5.2.1.301 EventExposureEnd

`quickSpinIntegerNode` EventExposureEnd

#### 5.2.1.302 EventExposureEndFrameID

`quickSpinIntegerNode` EventExposureEndFrameID

#### 5.2.1.303 EventExposureEndTimestamp

`quickSpinIntegerNode` EventExposureEndTimestamp

#### 5.2.1.304 EventExposureStart

`quickSpinIntegerNode` EventExposureStart

#### 5.2.1.305 EventExposureStartFrameID

`quickSpinIntegerNode` EventExposureStartFrameID

#### 5.2.1.306 EventExposureStartTimestamp

`quickSpinIntegerNode` EventExposureStartTimestamp

#### 5.2.1.307 EventFrameBurstEnd

`quickSpinIntegerNode` EventFrameBurstEnd

#### 5.2.1.308 EventFrameBurstEndFrameID

`quickSpinIntegerNode` EventFrameBurstEndFrameID

#### 5.2.1.309 EventFrameBurstEndTimestamp

[quickSpinIntegerNode](#) EventFrameBurstEndTimestamp

#### 5.2.1.310 EventFrameBurstStart

[quickSpinIntegerNode](#) EventFrameBurstStart

#### 5.2.1.311 EventFrameBurstStartFrameID

[quickSpinIntegerNode](#) EventFrameBurstStartFrameID

#### 5.2.1.312 EventFrameBurstStartTimestamp

[quickSpinIntegerNode](#) EventFrameBurstStartTimestamp

#### 5.2.1.313 EventFrameEnd

[quickSpinIntegerNode](#) EventFrameEnd

#### 5.2.1.314 EventFrameEndFrameID

[quickSpinIntegerNode](#) EventFrameEndFrameID

#### 5.2.1.315 EventFrameEndTimestamp

[quickSpinIntegerNode](#) EventFrameEndTimestamp

#### 5.2.1.316 EventFrameStart

[quickSpinIntegerNode](#) EventFrameStart

**5.2.1.317 EventFrameStartFrameID**

`quickSpinIntegerNode` EventFrameStartFrameID

**5.2.1.318 EventFrameStartTimestamp**

`quickSpinIntegerNode` EventFrameStartTimestamp

**5.2.1.319 EventFrameTransferEnd**

`quickSpinIntegerNode` EventFrameTransferEnd

**5.2.1.320 EventFrameTransferEndFrameID**

`quickSpinIntegerNode` EventFrameTransferEndFrameID

**5.2.1.321 EventFrameTransferEndTimestamp**

`quickSpinIntegerNode` EventFrameTransferEndTimestamp

**5.2.1.322 EventFrameTransferStart**

`quickSpinIntegerNode` EventFrameTransferStart

**5.2.1.323 EventFrameTransferStartFrameID**

`quickSpinIntegerNode` EventFrameTransferStartFrameID

**5.2.1.324 EventFrameTransferStartTimestamp**

`quickSpinIntegerNode` EventFrameTransferStartTimestamp



#### 5.2.1.325 EventFrameTrigger

`quickSpinIntegerNode` EventFrameTrigger

#### 5.2.1.326 EventFrameTriggerFrameID

`quickSpinIntegerNode` EventFrameTriggerFrameID

#### 5.2.1.327 EventFrameTriggerTimestamp

`quickSpinIntegerNode` EventFrameTriggerTimestamp

#### 5.2.1.328 EventLine0AnyEdge

`quickSpinIntegerNode` EventLine0AnyEdge

#### 5.2.1.329 EventLine0AnyEdgeFrameID

`quickSpinIntegerNode` EventLine0AnyEdgeFrameID

#### 5.2.1.330 EventLine0AnyEdgeTimestamp

`quickSpinIntegerNode` EventLine0AnyEdgeTimestamp

#### 5.2.1.331 EventLine0FallingEdge

`quickSpinIntegerNode` EventLine0FallingEdge

#### 5.2.1.332 EventLine0FallingEdgeFrameID

`quickSpinIntegerNode` EventLine0FallingEdgeFrameID

#### 5.2.1.333 EventLine0FallingEdgeTimestamp

`quickSpinIntegerNode` EventLine0FallingEdgeTimestamp

#### 5.2.1.334 EventLine0RisingEdge

`quickSpinIntegerNode` EventLine0RisingEdge

#### 5.2.1.335 EventLine0RisingEdgeFrameID

`quickSpinIntegerNode` EventLine0RisingEdgeFrameID

#### 5.2.1.336 EventLine0RisingEdgeTimestamp

`quickSpinIntegerNode` EventLine0RisingEdgeTimestamp

#### 5.2.1.337 EventLine1AnyEdge

`quickSpinIntegerNode` EventLine1AnyEdge

#### 5.2.1.338 EventLine1AnyEdgeFrameID

`quickSpinIntegerNode` EventLine1AnyEdgeFrameID

#### 5.2.1.339 EventLine1AnyEdgeTimestamp

`quickSpinIntegerNode` EventLine1AnyEdgeTimestamp

#### 5.2.1.340 EventLine1FallingEdge

`quickSpinIntegerNode` EventLine1FallingEdge

**5.2.1.341 EventLine1FallingEdgeFrameID**

`quickSpinIntegerNode` EventLine1FallingEdgeFrameID

**5.2.1.342 EventLine1FallingEdgeTimestamp**

`quickSpinIntegerNode` EventLine1FallingEdgeTimestamp

**5.2.1.343 EventLine1RisingEdge**

`quickSpinIntegerNode` EventLine1RisingEdge

**5.2.1.344 EventLine1RisingEdgeFrameID**

`quickSpinIntegerNode` EventLine1RisingEdgeFrameID

**5.2.1.345 EventLine1RisingEdgeTimestamp**

`quickSpinIntegerNode` EventLine1RisingEdgeTimestamp

**5.2.1.346 EventLinkSpeedChange**

`quickSpinIntegerNode` EventLinkSpeedChange

**5.2.1.347 EventLinkSpeedChangeFrameID**

`quickSpinIntegerNode` EventLinkSpeedChangeFrameID

**5.2.1.348 EventLinkSpeedChangeTimestamp**

`quickSpinIntegerNode` EventLinkSpeedChangeTimestamp

#### 5.2.1.349 EventLinkTrigger0

`quickSpinIntegerNode` EventLinkTrigger0

#### 5.2.1.350 EventLinkTrigger0FrameID

`quickSpinIntegerNode` EventLinkTrigger0FrameID

#### 5.2.1.351 EventLinkTrigger0Timestamp

`quickSpinIntegerNode` EventLinkTrigger0Timestamp

#### 5.2.1.352 EventLinkTrigger1

`quickSpinIntegerNode` EventLinkTrigger1

#### 5.2.1.353 EventLinkTrigger1FrameID

`quickSpinIntegerNode` EventLinkTrigger1FrameID

#### 5.2.1.354 EventLinkTrigger1Timestamp

`quickSpinIntegerNode` EventLinkTrigger1Timestamp

#### 5.2.1.355 EventNotification

`quickSpinEnumerationNode` EventNotification

#### 5.2.1.356 EventSelector

`quickSpinEnumerationNode` EventSelector

#### 5.2.1.357 EventSequencerSetChange

[quickSpinIntegerNode](#) EventSequencerSetChange

#### 5.2.1.358 EventSequencerSetChangeFrameID

[quickSpinIntegerNode](#) EventSequencerSetChangeFrameID

#### 5.2.1.359 EventSequencerSetChangeTimestamp

[quickSpinIntegerNode](#) EventSequencerSetChangeTimestamp

#### 5.2.1.360 EventSerialData

[quickSpinStringNode](#) EventSerialData

#### 5.2.1.361 EventSerialDataLength

[quickSpinIntegerNode](#) EventSerialDataLength

#### 5.2.1.362 EventSerialPortReceive

[quickSpinIntegerNode](#) EventSerialPortReceive

#### 5.2.1.363 EventSerialPortReceiveTimestamp

[quickSpinIntegerNode](#) EventSerialPortReceiveTimestamp

#### 5.2.1.364 EventSerialReceiveOverflow

[quickSpinBooleanNode](#) EventSerialReceiveOverflow

#### 5.2.1.365 EventStream0TransferBlockEnd

`quickSpinIntegerNode` EventStream0TransferBlockEnd

#### 5.2.1.366 EventStream0TransferBlockEndFrameID

`quickSpinIntegerNode` EventStream0TransferBlockEndFrameID

#### 5.2.1.367 EventStream0TransferBlockEndTimestamp

`quickSpinIntegerNode` EventStream0TransferBlockEndTimestamp

#### 5.2.1.368 EventStream0TransferBlockStart

`quickSpinIntegerNode` EventStream0TransferBlockStart

#### 5.2.1.369 EventStream0TransferBlockStartFrameID

`quickSpinIntegerNode` EventStream0TransferBlockStartFrameID

#### 5.2.1.370 EventStream0TransferBlockStartTimestamp

`quickSpinIntegerNode` EventStream0TransferBlockStartTimestamp

#### 5.2.1.371 EventStream0TransferBlockTrigger

`quickSpinIntegerNode` EventStream0TransferBlockTrigger

#### 5.2.1.372 EventStream0TransferBlockTriggerFrameID

`quickSpinIntegerNode` EventStream0TransferBlockTriggerFrameID

**5.2.1.373 EventStream0TransferBlockTriggerTimestamp**

`quickSpinIntegerNode` EventStream0TransferBlockTriggerTimestamp

**5.2.1.374 EventStream0TransferBurstEnd**

`quickSpinIntegerNode` EventStream0TransferBurstEnd

**5.2.1.375 EventStream0TransferBurstEndFrameID**

`quickSpinIntegerNode` EventStream0TransferBurstEndFrameID

**5.2.1.376 EventStream0TransferBurstEndTimestamp**

`quickSpinIntegerNode` EventStream0TransferBurstEndTimestamp

**5.2.1.377 EventStream0TransferBurstStart**

`quickSpinIntegerNode` EventStream0TransferBurstStart

**5.2.1.378 EventStream0TransferBurstStartFrameID**

`quickSpinIntegerNode` EventStream0TransferBurstStartFrameID

**5.2.1.379 EventStream0TransferBurstStartTimestamp**

`quickSpinIntegerNode` EventStream0TransferBurstStartTimestamp

**5.2.1.380 EventStream0TransferEnd**

`quickSpinIntegerNode` EventStream0TransferEnd

**5.2.1.381 EventStream0TransferEndFrameID**

`quickSpinIntegerNode` EventStream0TransferEndFrameID

**5.2.1.382 EventStream0TransferEndTimestamp**

`quickSpinIntegerNode` EventStream0TransferEndTimestamp

**5.2.1.383 EventStream0TransferOverflow**

`quickSpinIntegerNode` EventStream0TransferOverflow

**5.2.1.384 EventStream0TransferOverflowFrameID**

`quickSpinIntegerNode` EventStream0TransferOverflowFrameID

**5.2.1.385 EventStream0TransferOverflowTimestamp**

`quickSpinIntegerNode` EventStream0TransferOverflowTimestamp

**5.2.1.386 EventStream0TransferPause**

`quickSpinIntegerNode` EventStream0TransferPause

**5.2.1.387 EventStream0TransferPauseFrameID**

`quickSpinIntegerNode` EventStream0TransferPauseFrameID

**5.2.1.388 EventStream0TransferPauseTimestamp**

`quickSpinIntegerNode` EventStream0TransferPauseTimestamp



**5.2.1.389 EventStream0TransferResume**

`quickSpinIntegerNode` EventStream0TransferResume

**5.2.1.390 EventStream0TransferResumeFrameID**

`quickSpinIntegerNode` EventStream0TransferResumeFrameID

**5.2.1.391 EventStream0TransferResumeTimestamp**

`quickSpinIntegerNode` EventStream0TransferResumeTimestamp

**5.2.1.392 EventStream0TransferStart**

`quickSpinIntegerNode` EventStream0TransferStart

**5.2.1.393 EventStream0TransferStartFrameID**

`quickSpinIntegerNode` EventStream0TransferStartFrameID

**5.2.1.394 EventStream0TransferStartTimestamp**

`quickSpinIntegerNode` EventStream0TransferStartTimestamp

**5.2.1.395 EventTest**

`quickSpinIntegerNode` EventTest

**5.2.1.396 EventTestTimestamp**

`quickSpinIntegerNode` EventTestTimestamp

**5.2.1.397 EventTimer0End**

`quickSpinIntegerNode` EventTimer0End

**5.2.1.398 EventTimer0EndFrameID**

`quickSpinIntegerNode` EventTimer0EndFrameID

**5.2.1.399 EventTimer0EndTimestamp**

`quickSpinIntegerNode` EventTimer0EndTimestamp

**5.2.1.400 EventTimer0Start**

`quickSpinIntegerNode` EventTimer0Start

**5.2.1.401 EventTimer0StartFrameID**

`quickSpinIntegerNode` EventTimer0StartFrameID

**5.2.1.402 EventTimer0StartTimestamp**

`quickSpinIntegerNode` EventTimer0StartTimestamp

**5.2.1.403 EventTimer1End**

`quickSpinIntegerNode` EventTimer1End

**5.2.1.404 EventTimer1EndFrameID**

`quickSpinIntegerNode` EventTimer1EndFrameID

#### 5.2.1.405 EventTimer1EndTimestamp

`quickSpinIntegerNode` EventTimer1EndTimestamp

#### 5.2.1.406 EventTimer1Start

`quickSpinIntegerNode` EventTimer1Start

#### 5.2.1.407 EventTimer1StartFrameID

`quickSpinIntegerNode` EventTimer1StartFrameID

#### 5.2.1.408 EventTimer1StartTimestamp

`quickSpinIntegerNode` EventTimer1StartTimestamp

#### 5.2.1.409 ExposureActiveMode

`quickSpinEnumerationNode` ExposureActiveMode

#### 5.2.1.410 ExposureAuto

`quickSpinEnumerationNode` ExposureAuto

#### 5.2.1.411 ExposureMode

`quickSpinEnumerationNode` ExposureMode

#### 5.2.1.412 ExposureTime

`quickSpinFloatNode` ExposureTime

**5.2.1.413 ExposureTimeMode**

[quickSpinEnumerationNode](#) ExposureTimeMode

**5.2.1.414 ExposureTimeSelector**

[quickSpinEnumerationNode](#) ExposureTimeSelector

**5.2.1.415 FactoryReset**

[quickSpinCommandNode](#) FactoryReset

**5.2.1.416 FileAccessBuffer**

[quickSpinRegisterNode](#) FileAccessBuffer

**5.2.1.417 FileAccessLength**

[quickSpinIntegerNode](#) FileAccessLength

**5.2.1.418 FileAccessOffset**

[quickSpinIntegerNode](#) FileAccessOffset

**5.2.1.419 FileOpenMode**

[quickSpinEnumerationNode](#) FileOpenMode

**5.2.1.420 FileOperationExecute**

[quickSpinCommandNode](#) FileOperationExecute

**5.2.1.421 FileOperationResult**

`quickSpinIntegerNode` FileOperationResult

**5.2.1.422 FileOperationSelector**

`quickSpinEnumerationNode` FileOperationSelector

**5.2.1.423 FileOperationStatus**

`quickSpinEnumerationNode` FileOperationStatus

**5.2.1.424 FileSelector**

`quickSpinEnumerationNode` FileSelector

**5.2.1.425 FileSize**

`quickSpinIntegerNode` FileSize

**5.2.1.426 Gain**

`quickSpinFloatNode` Gain

**5.2.1.427 GainAuto**

`quickSpinEnumerationNode` GainAuto

**5.2.1.428 GainAutoBalance**

`quickSpinEnumerationNode` GainAutoBalance

**5.2.1.429 GainSelector**

[quickSpinEnumerationNode](#) GainSelector

**5.2.1.430 Gamma**

[quickSpinFloatNode](#) Gamma

**5.2.1.431 GammaEnable**

[quickSpinBooleanNode](#) GammaEnable

**5.2.1.432 GevActiveLinkCount**

[quickSpinIntegerNode](#) GevActiveLinkCount

**5.2.1.433 GevCCP**

[quickSpinEnumerationNode](#) GevCCP

**5.2.1.434 GevCurrentDefaultGateway**

[quickSpinIntegerNode](#) GevCurrentDefaultGateway

**5.2.1.435 GevCurrentIPAddress**

[quickSpinIntegerNode](#) GevCurrentIPAddress

**5.2.1.436 GevCurrentIPConfigurationDHCP**

[quickSpinBooleanNode](#) GevCurrentIPConfigurationDHCP

**5.2.1.437   GevCurrentIPConfigurationLLA**

`quickSpinBooleanNode`   GevCurrentIPConfigurationLLA

**5.2.1.438   GevCurrentIPConfigurationPersistentIP**

`quickSpinBooleanNode`   GevCurrentIPConfigurationPersistentIP

**5.2.1.439   GevCurrentPhysicalLinkConfiguration**

`quickSpinEnumerationNode`   GevCurrentPhysicalLinkConfiguration

**5.2.1.440   GevCurrentSubnetMask**

`quickSpinIntegerNode`   GevCurrentSubnetMask

**5.2.1.441   GevDiscoveryAckDelay**

`quickSpinIntegerNode`   GevDiscoveryAckDelay

**5.2.1.442   GevFirstURL**

`quickSpinStringNode`   GevFirstURL

**5.2.1.443   GevGVCPExtendedStatusCodes**

`quickSpinBooleanNode`   GevGVCPExtendedStatusCodes

**5.2.1.444   GevGVCPExtendedStatusCodesSelector**

`quickSpinEnumerationNode`   GevGVCPExtendedStatusCodesSelector

#### 5.2.1.445 **GevGVCPHeartbeatDisable**

`quickSpinBooleanNode` `GevGVCPHeartbeatDisable`

#### 5.2.1.446 **GevGVCPPendingAck**

`quickSpinBooleanNode` `GevGVCPPendingAck`

#### 5.2.1.447 **GevGVCPPendingTimeout**

`quickSpinIntegerNode` `GevGVCPPendingTimeout`

#### 5.2.1.448 **GevGVSPExtendedIDMode**

`quickSpinEnumerationNode` `GevGVSPExtendedIDMode`

#### 5.2.1.449 **GevHeartbeatTimeout**

`quickSpinIntegerNode` `GevHeartbeatTimeout`

#### 5.2.1.450 **GevIEEE1588**

`quickSpinBooleanNode` `GevIEEE1588`

#### 5.2.1.451 **GevIEEE1588ClockAccuracy**

`quickSpinEnumerationNode` `GevIEEE1588ClockAccuracy`

#### 5.2.1.452 **GevIEEE1588Mode**

`quickSpinEnumerationNode` `GevIEEE1588Mode`



**5.2.1.453   GevIEEE1588Status**

`quickSpinEnumerationNode` `GevIEEE1588Status`

**5.2.1.454   GevInterfaceSelector**

`quickSpinIntegerNode` `GevInterfaceSelector`

**5.2.1.455   GevIPConfigurationStatus**

`quickSpinEnumerationNode` `GevIPConfigurationStatus`

**5.2.1.456   GevMACAddress**

`quickSpinIntegerNode` `GevMACAddress`

**5.2.1.457   GevMCDA**

`quickSpinIntegerNode` `GevMCDA`

**5.2.1.458   GevMCPHostPort**

`quickSpinIntegerNode` `GevMCPHostPort`

**5.2.1.459   GevMCRC**

`quickSpinIntegerNode` `GevMCRC`

**5.2.1.460   GevMCSP**

`quickSpinIntegerNode` `GevMCSP`

**5.2.1.461   GevMCTT**

`quickSpinIntegerNode`   GevMCTT

**5.2.1.462   GevNumberOfInterfaces**

`quickSpinIntegerNode`   GevNumberOfInterfaces

**5.2.1.463   GevPAUSEFrameReception**

`quickSpinBooleanNode`   GevPAUSEFrameReception

**5.2.1.464   GevPAUSEFrameTransmission**

`quickSpinBooleanNode`   GevPAUSEFrameTransmission

**5.2.1.465   GevPersistentDefaultGateway**

`quickSpinIntegerNode`   GevPersistentDefaultGateway

**5.2.1.466   GevPersistentIPAddress**

`quickSpinIntegerNode`   GevPersistentIPAddress

**5.2.1.467   GevPersistentSubnetMask**

`quickSpinIntegerNode`   GevPersistentSubnetMask

**5.2.1.468   GevPhysicalLinkConfiguration**

`quickSpinEnumerationNode`   GevPhysicalLinkConfiguration

**5.2.1.469   GevPrimaryApplicationIPAddress**

`quickSpinIntegerNode`   GevPrimaryApplicationIPAddress

**5.2.1.470   GevPrimaryApplicationSocket**

`quickSpinIntegerNode`   GevPrimaryApplicationSocket

**5.2.1.471   GevPrimaryApplicationSwitchoverKey**

`quickSpinIntegerNode`   GevPrimaryApplicationSwitchoverKey

**5.2.1.472   GevSCCFGAllInTransmission**

`quickSpinBooleanNode`   GevSCCFGAllInTransmission

**5.2.1.473   GevSCCFGExtendedChunkData**

`quickSpinBooleanNode`   GevSCCFGExtendedChunkData

**5.2.1.474   GevSCCFGPacketResendDestination**

`quickSpinBooleanNode`   GevSCCFGPacketResendDestination

**5.2.1.475   GevSCCFGUnconditionalStreaming**

`quickSpinBooleanNode`   GevSCCFGUnconditionalStreaming

**5.2.1.476   GevSCDA**

`quickSpinIntegerNode`   GevSCDA

**5.2.1.477   GevSCPD**

[quickSpinIntegerNode](#)   GevSCPD

**5.2.1.478   GevSCPDirection**

[quickSpinIntegerNode](#)   GevSCPDirection

**5.2.1.479   GevSCPHostPort**

[quickSpinIntegerNode](#)   GevSCPHostPort

**5.2.1.480   GevSCPInterfaceIndex**

[quickSpinIntegerNode](#)   GevSCPInterfaceIndex

**5.2.1.481   GevSCPSBigEndian**

[quickSpinBooleanNode](#)   GevSCPSBigEndian

**5.2.1.482   GevSCPSDoNotFragment**

[quickSpinBooleanNode](#)   GevSCPSDoNotFragment

**5.2.1.483   GevSCPSFireTestPacket**

[quickSpinBooleanNode](#)   GevSCPSFireTestPacket

**5.2.1.484   GevSCPSPacketSize**

[quickSpinIntegerNode](#)   GevSCPSPacketSize

**5.2.1.485   GevSCSP**

[quickSpinIntegerNode](#)   GevSCSP

**5.2.1.486   GevSCZoneConfigurationLock**

[quickSpinBooleanNode](#)   GevSCZoneConfigurationLock

**5.2.1.487   GevSCZoneCount**

[quickSpinIntegerNode](#)   GevSCZoneCount

**5.2.1.488   GevSCZoneDirectionAll**

[quickSpinIntegerNode](#)   GevSCZoneDirectionAll

**5.2.1.489   GevSecondURL**

[quickSpinStringNode](#)   GevSecondURL

**5.2.1.490   GevStreamChannelSelector**

[quickSpinIntegerNode](#)   GevStreamChannelSelector

**5.2.1.491   GevSupportedOption**

[quickSpinBooleanNode](#)   GevSupportedOption

**5.2.1.492   GevSupportedOptionSelector**

[quickSpinEnumerationNode](#)   GevSupportedOptionSelector

**5.2.1.493   GevTimestampTickFrequency**

`quickSpinIntegerNode`   GevTimestampTickFrequency

**5.2.1.494   GuiXmlManifestAddress**

`quickSpinIntegerNode`   GuiXmlManifestAddress

**5.2.1.495   Height**

`quickSpinIntegerNode`   Height

**5.2.1.496   HeightMax**

`quickSpinIntegerNode`   HeightMax

**5.2.1.497   ImageComponentEnable**

`quickSpinBooleanNode`   ImageComponentEnable

**5.2.1.498   ImageComponentSelector**

`quickSpinEnumerationNode`   ImageComponentSelector

**5.2.1.499   ImageCompressionBitrate**

`quickSpinFloatNode`   ImageCompressionBitrate

**5.2.1.500   ImageCompressionJPEGFormatOption**

`quickSpinEnumerationNode`   ImageCompressionJPEGFormatOption

#### 5.2.1.501 ImageCompressionMode

`quickSpinEnumerationNode` ImageCompressionMode

#### 5.2.1.502 ImageCompressionQuality

`quickSpinIntegerNode` ImageCompressionQuality

#### 5.2.1.503 ImageCompressionRateOption

`quickSpinEnumerationNode` ImageCompressionRateOption

#### 5.2.1.504 IspEnable

`quickSpinBooleanNode` IspEnable

#### 5.2.1.505 LineFilterWidth

`quickSpinFloatNode` LineFilterWidth

#### 5.2.1.506 LineFormat

`quickSpinEnumerationNode` LineFormat

#### 5.2.1.507 LineInputFilterSelector

`quickSpinEnumerationNode` LineInputFilterSelector

#### 5.2.1.508 LineInverter

`quickSpinBooleanNode` LineInverter

**5.2.1.509 LineMode**

`quickSpinEnumerationNode` LineMode

**5.2.1.510 LinePitch**

`quickSpinIntegerNode` LinePitch

**5.2.1.511 LineSelector**

`quickSpinEnumerationNode` LineSelector

**5.2.1.512 LineSource**

`quickSpinEnumerationNode` LineSource

**5.2.1.513 LineStatus**

`quickSpinBooleanNode` LineStatus

**5.2.1.514 LineStatusAll**

`quickSpinIntegerNode` LineStatusAll

**5.2.1.515 LinkErrorCount**

`quickSpinIntegerNode` LinkErrorCount

**5.2.1.516 LinkUptime**

`quickSpinIntegerNode` LinkUptime



**5.2.1.517 LogicBlockLUTInputActivation**

`quickSpinEnumerationNode` LogicBlockLUTInputActivation

**5.2.1.518 LogicBlockLUTInputSelector**

`quickSpinEnumerationNode` LogicBlockLUTInputSelector

**5.2.1.519 LogicBlockLUTInputSource**

`quickSpinEnumerationNode` LogicBlockLUTInputSource

**5.2.1.520 LogicBlockLUTOutputValue**

`quickSpinBooleanNode` LogicBlockLUTOutputValue

**5.2.1.521 LogicBlockLUTOutputValueAll**

`quickSpinIntegerNode` LogicBlockLUTOutputValueAll

**5.2.1.522 LogicBlockLUTRowIndex**

`quickSpinIntegerNode` LogicBlockLUTRowIndex

**5.2.1.523 LogicBlockLUTSelector**

`quickSpinEnumerationNode` LogicBlockLUTSelector

**5.2.1.524 LogicBlockSelector**

`quickSpinEnumerationNode` LogicBlockSelector

**5.2.1.525 LUTEnable**

`quickSpinBooleanNode` LUTEnable

**5.2.1.526 LUTIndex**

`quickSpinIntegerNode` LUTIndex

**5.2.1.527 LUTSelector**

`quickSpinEnumerationNode` LUTSelector

**5.2.1.528 LUTValue**

`quickSpinIntegerNode` LUTValue

**5.2.1.529 LUTValueAll**

`quickSpinRegisterNode` LUTValueAll

**5.2.1.530 MaxDeviceResetTime**

`quickSpinIntegerNode` MaxDeviceResetTime

**5.2.1.531 OffsetX**

`quickSpinIntegerNode` OffsetX

**5.2.1.532 OffsetY**

`quickSpinIntegerNode` OffsetY

#### 5.2.1.533 PacketResendRequestCount

`quickSpinIntegerNode` PacketResendRequestCount

#### 5.2.1.534 PayloadSize

`quickSpinIntegerNode` PayloadSize

#### 5.2.1.535 PixelColorFilter

`quickSpinEnumerationNode` PixelColorFilter

#### 5.2.1.536 PixelDynamicRangeMax

`quickSpinIntegerNode` PixelDynamicRangeMax

#### 5.2.1.537 PixelDynamicRangeMin

`quickSpinIntegerNode` PixelDynamicRangeMin

#### 5.2.1.538 PixelFormat

`quickSpinEnumerationNode` PixelFormat

#### 5.2.1.539 PixelFormatInfoID

`quickSpinIntegerNode` PixelFormatInfoID

#### 5.2.1.540 PixelFormatInfoSelector

`quickSpinEnumerationNode` PixelFormatInfoSelector

**5.2.1.541 PixelSize**

`quickSpinEnumerationNode` PixelSize

**5.2.1.542 PowerSupplyCurrent**

`quickSpinFloatNode` PowerSupplyCurrent

**5.2.1.543 PowerSupplyVoltage**

`quickSpinFloatNode` PowerSupplyVoltage

**5.2.1.544 RegionDestination**

`quickSpinEnumerationNode` RegionDestination

**5.2.1.545 RegionMode**

`quickSpinEnumerationNode` RegionMode

**5.2.1.546 RegionSelector**

`quickSpinEnumerationNode` RegionSelector

**5.2.1.547 ReverseX**

`quickSpinBooleanNode` ReverseX

**5.2.1.548 ReverseY**

`quickSpinBooleanNode` ReverseY

#### 5.2.1.549 RgbTransformLightSource

`quickSpinEnumerationNode` RgbTransformLightSource

#### 5.2.1.550 Saturation

`quickSpinFloatNode` Saturation

#### 5.2.1.551 SaturationEnable

`quickSpinBooleanNode` SaturationEnable

#### 5.2.1.552 Scan3dAxisMax

`quickSpinFloatNode` Scan3dAxisMax

#### 5.2.1.553 Scan3dAxisMin

`quickSpinFloatNode` Scan3dAxisMin

#### 5.2.1.554 Scan3dCoordinateOffset

`quickSpinFloatNode` Scan3dCoordinateOffset

#### 5.2.1.555 Scan3dCoordinateReferenceSelector

`quickSpinEnumerationNode` Scan3dCoordinateReferenceSelector

#### 5.2.1.556 Scan3dCoordinateReferenceValue

`quickSpinFloatNode` Scan3dCoordinateReferenceValue

**5.2.1.557 Scan3dCoordinateScale**

`quickSpinFloatNode` `Scan3dCoordinateScale`

**5.2.1.558 Scan3dCoordinateSelector**

`quickSpinEnumerationNode` `Scan3dCoordinateSelector`

**5.2.1.559 Scan3dCoordinateSystem**

`quickSpinEnumerationNode` `Scan3dCoordinateSystem`

**5.2.1.560 Scan3dCoordinateSystemReference**

`quickSpinEnumerationNode` `Scan3dCoordinateSystemReference`

**5.2.1.561 Scan3dCoordinateTransformSelector**

`quickSpinEnumerationNode` `Scan3dCoordinateTransformSelector`

**5.2.1.562 Scan3dDistanceUnit**

`quickSpinEnumerationNode` `Scan3dDistanceUnit`

**5.2.1.563 Scan3dInvalidDataFlag**

`quickSpinBooleanNode` `Scan3dInvalidDataFlag`

**5.2.1.564 Scan3dInvalidDataValue**

`quickSpinFloatNode` `Scan3dInvalidDataValue`

**5.2.1.565 Scan3dOutputMode**

`quickSpinEnumerationNode` Scan3dOutputMode

**5.2.1.566 Scan3dTransformValue**

`quickSpinFloatNode` Scan3dTransformValue

**5.2.1.567 SensorDescription**

`quickSpinStringNode` SensorDescription

**5.2.1.568 SensorDigitizationTaps**

`quickSpinEnumerationNode` SensorDigitizationTaps

**5.2.1.569 SensorHeight**

`quickSpinIntegerNode` SensorHeight

**5.2.1.570 SensorShutterMode**

`quickSpinEnumerationNode` SensorShutterMode

**5.2.1.571 SensorTaps**

`quickSpinEnumerationNode` SensorTaps

**5.2.1.572 SensorWidth**

`quickSpinIntegerNode` SensorWidth

#### 5.2.1.573 SequencerConfigurationMode

[quickSpinEnumerationNode](#) SequencerConfigurationMode

#### 5.2.1.574 SequencerConfigurationValid

[quickSpinEnumerationNode](#) SequencerConfigurationValid

#### 5.2.1.575 SequencerFeatureEnable

[quickSpinBooleanNode](#) SequencerFeatureEnable

#### 5.2.1.576 SequencerMode

[quickSpinEnumerationNode](#) SequencerMode

#### 5.2.1.577 SequencerPathSelector

[quickSpinIntegerNode](#) SequencerPathSelector

#### 5.2.1.578 SequencerSetActive

[quickSpinIntegerNode](#) SequencerSetActive

#### 5.2.1.579 SequencerSetLoad

[quickSpinCommandNode](#) SequencerSetLoad

#### 5.2.1.580 SequencerSetNext

[quickSpinIntegerNode](#) SequencerSetNext



**5.2.1.581 SequencerSetSave**

[quickSpinCommandNode](#) SequencerSetSave

**5.2.1.582 SequencerSetSelector**

[quickSpinIntegerNode](#) SequencerSetSelector

**5.2.1.583 SequencerSetStart**

[quickSpinIntegerNode](#) SequencerSetStart

**5.2.1.584 SequencerSetValid**

[quickSpinEnumerationNode](#) SequencerSetValid

**5.2.1.585 SequencerTriggerActivation**

[quickSpinEnumerationNode](#) SequencerTriggerActivation

**5.2.1.586 SequencerTriggerSource**

[quickSpinEnumerationNode](#) SequencerTriggerSource

**5.2.1.587 SerialPortBaudRate**

[quickSpinEnumerationNode](#) SerialPortBaudRate

**5.2.1.588 SerialPortDataBits**

[quickSpinIntegerNode](#) SerialPortDataBits

**5.2.1.589 SerialPortParity**

`quickSpinEnumerationNode` SerialPortParity

**5.2.1.590 SerialPortSelector**

`quickSpinEnumerationNode` SerialPortSelector

**5.2.1.591 SerialPortSource**

`quickSpinEnumerationNode` SerialPortSource

**5.2.1.592 SerialPortStopBits**

`quickSpinEnumerationNode` SerialPortStopBits

**5.2.1.593 SerialReceiveFramingErrorCount**

`quickSpinIntegerNode` SerialReceiveFramingErrorCount

**5.2.1.594 SerialReceiveParityErrorCount**

`quickSpinIntegerNode` SerialReceiveParityErrorCount

**5.2.1.595 SerialReceiveQueueClear**

`quickSpinCommandNode` SerialReceiveQueueClear

**5.2.1.596 SerialReceiveQueueCurrentCharacterCount**

`quickSpinIntegerNode` SerialReceiveQueueCurrentCharacterCount

**5.2.1.597 SerialReceiveQueueMaxCharacterCount**

`quickSpinIntegerNode` SerialReceiveQueueMaxCharacterCount

**5.2.1.598 SerialTransmitQueueCurrentCharacterCount**

`quickSpinIntegerNode` SerialTransmitQueueCurrentCharacterCount

**5.2.1.599 SerialTransmitQueueMaxCharacterCount**

`quickSpinIntegerNode` SerialTransmitQueueMaxCharacterCount

**5.2.1.600 Sharpening**

`quickSpinFloatNode` Sharpening

**5.2.1.601 SharpeningAuto**

`quickSpinBooleanNode` SharpeningAuto

**5.2.1.602 SharpeningEnable**

`quickSpinBooleanNode` SharpeningEnable

**5.2.1.603 SharpeningThreshold**

`quickSpinFloatNode` SharpeningThreshold

**5.2.1.604 SoftwareSignalPulse**

`quickSpinCommandNode` SoftwareSignalPulse

**5.2.1.605 SoftwareSignalSelector**

`quickSpinEnumerationNode` SoftwareSignalSelector

**5.2.1.606 SourceCount**

`quickSpinIntegerNode` SourceCount

**5.2.1.607 SourceSelector**

`quickSpinEnumerationNode` SourceSelector

**5.2.1.608 Test0001**

`quickSpinIntegerNode` Test0001

**5.2.1.609 TestEventGenerate**

`quickSpinCommandNode` TestEventGenerate

**5.2.1.610 TestPattern**

`quickSpinEnumerationNode` TestPattern

**5.2.1.611 TestPatternGeneratorSelector**

`quickSpinEnumerationNode` TestPatternGeneratorSelector

**5.2.1.612 TestPendingAck**

`quickSpinIntegerNode` TestPendingAck

### 5.2.1.613 TimerDelay

`quickSpinFloatNode` TimerDelay

### 5.2.1.614 TimerDuration

`quickSpinFloatNode` TimerDuration

### 5.2.1.615 TimerReset

`quickSpinCommandNode` TimerReset

### 5.2.1.616 TimerSelector

`quickSpinEnumerationNode` TimerSelector

### 5.2.1.617 TimerStatus

`quickSpinEnumerationNode` TimerStatus

### 5.2.1.618 TimerTriggerActivation

`quickSpinEnumerationNode` TimerTriggerActivation

### 5.2.1.619 TimerTriggerSource

`quickSpinEnumerationNode` TimerTriggerSource

### 5.2.1.620 TimerValue

`quickSpinFloatNode` TimerValue

**5.2.1.621 Timestamp**

[quickSpinIntegerNode](#) Timestamp

**5.2.1.622 TimestampLatch**

[quickSpinCommandNode](#) TimestampLatch

**5.2.1.623 TimestampLatchValue**

[quickSpinIntegerNode](#) TimestampLatchValue

**5.2.1.624 TimestampReset**

[quickSpinCommandNode](#) TimestampReset

**5.2.1.625 TLParamsLocked**

[quickSpinIntegerNode](#) TLParamsLocked

**5.2.1.626 TransferAbort**

[quickSpinCommandNode](#) TransferAbort

**5.2.1.627 TransferBlockCount**

[quickSpinIntegerNode](#) TransferBlockCount

**5.2.1.628 TransferBurstCount**

[quickSpinIntegerNode](#) TransferBurstCount

#### 5.2.1.629 TransferComponentSelector

`quickSpinEnumerationNode` TransferComponentSelector

#### 5.2.1.630 TransferControlMode

`quickSpinEnumerationNode` TransferControlMode

#### 5.2.1.631 TransferOperationMode

`quickSpinEnumerationNode` TransferOperationMode

#### 5.2.1.632 TransferPause

`quickSpinCommandNode` TransferPause

#### 5.2.1.633 TransferQueueCurrentBlockCount

`quickSpinIntegerNode` TransferQueueCurrentBlockCount

#### 5.2.1.634 TransferQueueMaxBlockCount

`quickSpinIntegerNode` TransferQueueMaxBlockCount

#### 5.2.1.635 TransferQueueMode

`quickSpinEnumerationNode` TransferQueueMode

#### 5.2.1.636 TransferQueueOverflowCount

`quickSpinIntegerNode` TransferQueueOverflowCount

**5.2.1.637 TransferResume**

`quickSpinCommandNode` TransferResume

**5.2.1.638 TransferSelector**

`quickSpinEnumerationNode` TransferSelector

**5.2.1.639 TransferStart**

`quickSpinCommandNode` TransferStart

**5.2.1.640 TransferStatus**

`quickSpinBooleanNode` TransferStatus

**5.2.1.641 TransferStatusSelector**

`quickSpinEnumerationNode` TransferStatusSelector

**5.2.1.642 TransferStop**

`quickSpinCommandNode` TransferStop

**5.2.1.643 TransferStreamChannel**

`quickSpinIntegerNode` TransferStreamChannel

**5.2.1.644 TransferTriggerActivation**

`quickSpinEnumerationNode` TransferTriggerActivation



#### 5.2.1.645 TransferTriggerMode

`quickSpinEnumerationNode` TransferTriggerMode

#### 5.2.1.646 TransferTriggerSelector

`quickSpinEnumerationNode` TransferTriggerSelector

#### 5.2.1.647 TransferTriggerSource

`quickSpinEnumerationNode` TransferTriggerSource

#### 5.2.1.648 TriggerActivation

`quickSpinEnumerationNode` TriggerActivation

#### 5.2.1.649 TriggerDelay

`quickSpinFloatNode` TriggerDelay

#### 5.2.1.650 TriggerDivider

`quickSpinIntegerNode` TriggerDivider

#### 5.2.1.651 TriggerEventTest

`quickSpinCommandNode` TriggerEventTest

#### 5.2.1.652 TriggerMode

`quickSpinEnumerationNode` TriggerMode

**5.2.1.653 TriggerMultiplier**

`quickSpinIntegerNode` TriggerMultiplier

**5.2.1.654 TriggerOverlap**

`quickSpinEnumerationNode` TriggerOverlap

**5.2.1.655 TriggerSelector**

`quickSpinEnumerationNode` TriggerSelector

**5.2.1.656 TriggerSoftware**

`quickSpinCommandNode` TriggerSoftware

**5.2.1.657 TriggerSource**

`quickSpinEnumerationNode` TriggerSource

**5.2.1.658 UserOutputSelector**

`quickSpinEnumerationNode` UserOutputSelector

**5.2.1.659 UserOutputValue**

`quickSpinBooleanNode` UserOutputValue

**5.2.1.660 UserOutputValueAll**

`quickSpinIntegerNode` UserOutputValueAll

**5.2.1.661 UserOutputValueAllMask**

`quickSpinIntegerNode` UserOutputValueAllMask

**5.2.1.662 UserSetDefault**

`quickSpinEnumerationNode` UserSetDefault

**5.2.1.663 UserSetFeatureEnable**

`quickSpinBooleanNode` UserSetFeatureEnable

**5.2.1.664 UserSetLoad**

`quickSpinCommandNode` UserSetLoad

**5.2.1.665 UserSetSave**

`quickSpinCommandNode` UserSetSave

**5.2.1.666 UserSetSelector**

`quickSpinEnumerationNode` UserSetSelector

**5.2.1.667 V3\_3Enable**

`quickSpinBooleanNode` V3\_3Enable

**5.2.1.668 WhiteClip**

`quickSpinFloatNode` WhiteClip

**5.2.1.669 WhiteClipSelector**

[quickSpinEnumerationNode](#) [WhiteClipSelector](#)

**5.2.1.670 Width**

[quickSpinIntegerNode](#) [Width](#)

**5.2.1.671 WidthMax**

[quickSpinIntegerNode](#) [WidthMax](#)

The documentation for this struct was generated from the following file:

- [include/spinc/QuickSpinDefsC.h](#)

**5.3 \_quickSpinTLDevice Struct Reference****Data Fields**

- [quickSpinStringNode](#) [DeviceID](#)
- [quickSpinStringNode](#) [DeviceSerialNumber](#)
- [quickSpinStringNode](#) [DeviceVendorName](#)
- [quickSpinStringNode](#) [DeviceModelName](#)
- [quickSpinEnumerationNode](#) [DeviceType](#)
- [quickSpinStringNode](#) [DeviceDisplayName](#)
- [quickSpinEnumerationNode](#) [DeviceAccessStatus](#)
- [quickSpinStringNode](#) [DeviceVersion](#)
- [quickSpinStringNode](#) [DeviceUserID](#)
- [quickSpinStringNode](#) [DeviceDriverVersion](#)
- [quickSpinBooleanNode](#) [DevicesUpdater](#)
- [quickSpinEnumerationNode](#) [GevCCP](#)
- [quickSpinEnumerationNode](#) [GUIXMLLocation](#)
- [quickSpinStringNode](#) [GUIXMLPath](#)
- [quickSpinEnumerationNode](#) [GenICamXMLLocation](#)
- [quickSpinStringNode](#) [GenICamXMLPath](#)
- [quickSpinIntegerNode](#) [GevDeviceIPAddress](#)
- [quickSpinIntegerNode](#) [GevDeviceSubnetMask](#)
- [quickSpinIntegerNode](#) [GevDeviceMACAddress](#)
- [quickSpinIntegerNode](#) [GevDeviceGateway](#)
- [quickSpinIntegerNode](#) [DeviceLinkSpeed](#)
- [quickSpinIntegerNode](#) [GevVersionMajor](#)
- [quickSpinIntegerNode](#) [GevVersionMinor](#)
- [quickSpinBooleanNode](#) [GevDeviceModelsBigEndian](#)
- [quickSpinIntegerNode](#) [GevDeviceReadAndWriteTimeout](#)

- [quickSpinIntegerNode](#) [GevDeviceMaximumRetryCount](#)
- [quickSpinIntegerNode](#) [GevDevicePort](#)
- [quickSpinCommandNode](#) [GevDeviceDiscoverMaximumPacketSize](#)
- [quickSpinIntegerNode](#) [GevDeviceMaximumPacketSize](#)
- [quickSpinBooleanNode](#) [GevDeviceIsWrongSubnet](#)
- [quickSpinCommandNode](#) [GevDeviceAutoForceIP](#)
- [quickSpinCommandNode](#) [GevDeviceForceIP](#)
- [quickSpinIntegerNode](#) [GevDeviceForceIPAddress](#)
- [quickSpinIntegerNode](#) [GevDeviceForceSubnetMask](#)
- [quickSpinIntegerNode](#) [GevDeviceForceGateway](#)
- [quickSpinBooleanNode](#) [DeviceMulticastMonitorMode](#)
- [quickSpinEnumerationNode](#) [DeviceEndiannessMechanism](#)
- [quickSpinStringNode](#) [DeviceInstanceId](#)
- [quickSpinStringNode](#) [DeviceLocation](#)
- [quickSpinEnumerationNode](#) [DeviceCurrentSpeed](#)
- [quickSpinBooleanNode](#) [DeviceU3VProtocol](#)

### 5.3.1 Field Documentation

#### 5.3.1.1 DeviceAccessStatus

[quickSpinEnumerationNode](#) [DeviceAccessStatus](#)

#### 5.3.1.2 DeviceCurrentSpeed

[quickSpinEnumerationNode](#) [DeviceCurrentSpeed](#)

#### 5.3.1.3 DeviceDisplayName

[quickSpinStringNode](#) [DeviceDisplayName](#)

#### 5.3.1.4 DeviceDriverVersion

[quickSpinStringNode](#) [DeviceDriverVersion](#)

#### 5.3.1.5 DeviceEndiannessMechanism

`quickSpinEnumerationNode` DeviceEndiannessMechanism

#### 5.3.1.6 DeviceID

`quickSpinStringNode` DeviceID

#### 5.3.1.7 DeviceInstanceId

`quickSpinStringNode` DeviceInstanceId

#### 5.3.1.8 DeviceIsUpdater

`quickSpinBooleanNode` DeviceIsUpdater

#### 5.3.1.9 DeviceLinkSpeed

`quickSpinIntegerNode` DeviceLinkSpeed

#### 5.3.1.10 DeviceLocation

`quickSpinStringNode` DeviceLocation

#### 5.3.1.11 DeviceModelName

`quickSpinStringNode` DeviceModelName

#### 5.3.1.12 DeviceMulticastMonitorMode

`quickSpinBooleanNode` DeviceMulticastMonitorMode

#### 5.3.1.13 DeviceSerialNumber

`quickSpinStringNode` DeviceSerialNumber

#### 5.3.1.14 DeviceType

`quickSpinEnumerationNode` DeviceType

#### 5.3.1.15 DeviceU3VProtocol

`quickSpinBooleanNode` DeviceU3VProtocol

#### 5.3.1.16 DeviceUserID

`quickSpinStringNode` DeviceUserID

#### 5.3.1.17 DeviceVendorName

`quickSpinStringNode` DeviceVendorName

#### 5.3.1.18 DeviceVersion

`quickSpinStringNode` DeviceVersion

#### 5.3.1.19 GenICamXMLLocation

`quickSpinEnumerationNode` GenICamXMLLocation

#### 5.3.1.20 GenICamXMLPath

`quickSpinStringNode` GenICamXMLPath

#### 5.3.1.21 **GevCCP**

`quickSpinEnumerationNode` `GevCCP`

#### 5.3.1.22 **GevDeviceAutoForceIP**

`quickSpinCommandNode` `GevDeviceAutoForceIP`

#### 5.3.1.23 **GevDeviceDiscoverMaximumPacketSize**

`quickSpinCommandNode` `GevDeviceDiscoverMaximumPacketSize`

#### 5.3.1.24 **GevDeviceForceGateway**

`quickSpinIntegerNode` `GevDeviceForceGateway`

#### 5.3.1.25 **GevDeviceForceIP**

`quickSpinCommandNode` `GevDeviceForceIP`

#### 5.3.1.26 **GevDeviceForceIPAddress**

`quickSpinIntegerNode` `GevDeviceForceIPAddress`

#### 5.3.1.27 **GevDeviceForceSubnetMask**

`quickSpinIntegerNode` `GevDeviceForceSubnetMask`

#### 5.3.1.28 **GevDeviceGateway**

`quickSpinIntegerNode` `GevDeviceGateway`



#### 5.3.1.29 GevDeviceIPAddress

`quickSpinIntegerNode` GevDeviceIPAddress

#### 5.3.1.30 GevDeviceIsWrongSubnet

`quickSpinBooleanNode` GevDeviceIsWrongSubnet

#### 5.3.1.31 GevDeviceMACAddress

`quickSpinIntegerNode` GevDeviceMACAddress

#### 5.3.1.32 GevDeviceMaximumPacketSize

`quickSpinIntegerNode` GevDeviceMaximumPacketSize

#### 5.3.1.33 GevDeviceMaximumRetryCount

`quickSpinIntegerNode` GevDeviceMaximumRetryCount

#### 5.3.1.34 GevDeviceModeIsBigEndian

`quickSpinBooleanNode` GevDeviceModeIsBigEndian

#### 5.3.1.35 GevDevicePort

`quickSpinIntegerNode` GevDevicePort

#### 5.3.1.36 GevDeviceReadAndWriteTimeout

`quickSpinIntegerNode` GevDeviceReadAndWriteTimeout

#### 5.3.1.37 **GevDeviceSubnetMask**

[quickSpinIntegerNode](#) [GevDeviceSubnetMask](#)

#### 5.3.1.38 **GevVersionMajor**

[quickSpinIntegerNode](#) [GevVersionMajor](#)

#### 5.3.1.39 **GevVersionMinor**

[quickSpinIntegerNode](#) [GevVersionMinor](#)

#### 5.3.1.40 **GUIXMLLocation**

[quickSpinEnumerationNode](#) [GUIXMLLocation](#)

#### 5.3.1.41 **GUIXMLPath**

[quickSpinStringNode](#) [GUIXMLPath](#)

The documentation for this struct was generated from the following file:

- [include/spinc/TransportLayerDeviceC.h](#)

## 5.4 **\_quickSpinTLInterface Struct Reference**

### Data Fields

- [quickSpinStringNode](#) [InterfaceID](#)
- [quickSpinStringNode](#) [InterfaceDisplayName](#)
- [quickSpinEnumerationNode](#) [InterfaceType](#)
- [quickSpinIntegerNode](#) [GevInterfaceGatewaySelector](#)
- [quickSpinIntegerNode](#) [GevInterfaceGateway](#)
- [quickSpinIntegerNode](#) [GevInterfaceMACAddress](#)
- [quickSpinIntegerNode](#) [GevInterfaceSubnetSelector](#)
- [quickSpinIntegerNode](#) [GevInterfaceSubnetIPAddress](#)
- [quickSpinIntegerNode](#) [GevInterfaceSubnetMask](#)
- [quickSpinIntegerNode](#) [GevInterfaceTransmitLinkSpeed](#)
- [quickSpinIntegerNode](#) [GevInterfaceReceiveLinkSpeed](#)

- [quickSpinIntegerNode](#) [GevInterfaceMTU](#)
- [quickSpinEnumerationNode](#) [POEStatus](#)
- [quickSpinEnumerationNode](#) [FilterDriverStatus](#)
- [quickSpinIntegerNode](#) [GevActionDeviceKey](#)
- [quickSpinIntegerNode](#) [GevActionGroupKey](#)
- [quickSpinIntegerNode](#) [GevActionGroupMask](#)
- [quickSpinIntegerNode](#) [GevActionTime](#)
- [quickSpinCommandNode](#) [ActionCommand](#)
- [quickSpinStringNode](#) [DeviceUnlock](#)
- [quickSpinCommandNode](#) [DeviceUpdateList](#)
- [quickSpinIntegerNode](#) [DeviceCount](#)
- [quickSpinIntegerNode](#) [DeviceSelector](#)
- [quickSpinStringNode](#) [DeviceID](#)
- [quickSpinStringNode](#) [DeviceVendorName](#)
- [quickSpinStringNode](#) [DeviceModelName](#)
- [quickSpinStringNode](#) [DeviceSerialNumber](#)
- [quickSpinEnumerationNode](#) [DeviceAccessStatus](#)
- [quickSpinIntegerNode](#) [GevDeviceIPAddress](#)
- [quickSpinIntegerNode](#) [GevDeviceSubnetMask](#)
- [quickSpinIntegerNode](#) [GevDeviceGateway](#)
- [quickSpinIntegerNode](#) [GevDeviceMACAddress](#)
- [quickSpinIntegerNode](#) [IncompatibleDeviceCount](#)
- [quickSpinIntegerNode](#) [IncompatibleDeviceSelector](#)
- [quickSpinStringNode](#) [IncompatibleDeviceID](#)
- [quickSpinStringNode](#) [IncompatibleDeviceVendorName](#)
- [quickSpinStringNode](#) [IncompatibleDeviceModelName](#)
- [quickSpinIntegerNode](#) [IncompatibleGevDeviceIPAddress](#)
- [quickSpinIntegerNode](#) [IncompatibleGevDeviceSubnetMask](#)
- [quickSpinIntegerNode](#) [IncompatibleGevDeviceMACAddress](#)
- [quickSpinCommandNode](#) [GevDeviceForceIP](#)
- [quickSpinIntegerNode](#) [GevDeviceForceIPAddress](#)
- [quickSpinIntegerNode](#) [GevDeviceForceSubnetMask](#)
- [quickSpinIntegerNode](#) [GevDeviceForceGateway](#)
- [quickSpinCommandNode](#) [GevDeviceAutoForceIP](#)
- [quickSpinStringNode](#) [HostAdapterName](#)
- [quickSpinStringNode](#) [HostAdapterVendor](#)
- [quickSpinStringNode](#) [HostAdapterDriverVersion](#)

## 5.4.1 Field Documentation

### 5.4.1.1 ActionCommand

[quickSpinCommandNode](#) [ActionCommand](#)

### 5.4.1.2 DeviceAccessStatus

[quickSpinEnumerationNode](#) [DeviceAccessStatus](#)

#### 5.4.1.3 DeviceCount

`quickSpinIntegerNode` DeviceCount

#### 5.4.1.4 DeviceID

`quickSpinStringNode` DeviceID

#### 5.4.1.5 DeviceModelName

`quickSpinStringNode` DeviceModelName

#### 5.4.1.6 DeviceSelector

`quickSpinIntegerNode` DeviceSelector

#### 5.4.1.7 DeviceSerialNumber

`quickSpinStringNode` DeviceSerialNumber

#### 5.4.1.8 DeviceUnlock

`quickSpinStringNode` DeviceUnlock

#### 5.4.1.9 DeviceUpdateList

`quickSpinCommandNode` DeviceUpdateList

#### 5.4.1.10 DeviceVendorName

`quickSpinStringNode` DeviceVendorName

#### 5.4.1.11 FilterDriverStatus

`quickSpinEnumerationNode` FilterDriverStatus

#### 5.4.1.12 GevActionDeviceKey

`quickSpinIntegerNode` GevActionDeviceKey

#### 5.4.1.13 GevActionGroupKey

`quickSpinIntegerNode` GevActionGroupKey

#### 5.4.1.14 GevActionGroupMask

`quickSpinIntegerNode` GevActionGroupMask

#### 5.4.1.15 GevActionTime

`quickSpinIntegerNode` GevActionTime

#### 5.4.1.16 GevDeviceAutoForceIP

`quickSpinCommandNode` GevDeviceAutoForceIP

#### 5.4.1.17 GevDeviceForceGateway

`quickSpinIntegerNode` GevDeviceForceGateway

#### 5.4.1.18 GevDeviceForceIP

`quickSpinCommandNode` GevDeviceForceIP

#### 5.4.1.19 **GevDeviceForceIPAddress**

`quickSpinIntegerNode` `GevDeviceForceIPAddress`

#### 5.4.1.20 **GevDeviceForceSubnetMask**

`quickSpinIntegerNode` `GevDeviceForceSubnetMask`

#### 5.4.1.21 **GevDeviceGateway**

`quickSpinIntegerNode` `GevDeviceGateway`

#### 5.4.1.22 **GevDeviceIPAddress**

`quickSpinIntegerNode` `GevDeviceIPAddress`

#### 5.4.1.23 **GevDeviceMACAddress**

`quickSpinIntegerNode` `GevDeviceMACAddress`

#### 5.4.1.24 **GevDeviceSubnetMask**

`quickSpinIntegerNode` `GevDeviceSubnetMask`

#### 5.4.1.25 **GevInterfaceGateway**

`quickSpinIntegerNode` `GevInterfaceGateway`

#### 5.4.1.26 **GevInterfaceGatewaySelector**

`quickSpinIntegerNode` `GevInterfaceGatewaySelector`

#### 5.4.1.27 **GevInterfaceMACAddress**

`quickSpinIntegerNode` `GevInterfaceMACAddress`

#### 5.4.1.28 **GevInterfaceMTU**

`quickSpinIntegerNode` `GevInterfaceMTU`

#### 5.4.1.29 **GevInterfaceReceiveLinkSpeed**

`quickSpinIntegerNode` `GevInterfaceReceiveLinkSpeed`

#### 5.4.1.30 **GevInterfaceSubnetIPAddress**

`quickSpinIntegerNode` `GevInterfaceSubnetIPAddress`

#### 5.4.1.31 **GevInterfaceSubnetMask**

`quickSpinIntegerNode` `GevInterfaceSubnetMask`

#### 5.4.1.32 **GevInterfaceSubnetSelector**

`quickSpinIntegerNode` `GevInterfaceSubnetSelector`

#### 5.4.1.33 **GevInterfaceTransmitLinkSpeed**

`quickSpinIntegerNode` `GevInterfaceTransmitLinkSpeed`

#### 5.4.1.34 **HostAdapterDriverVersion**

`quickSpinStringNode` `HostAdapterDriverVersion`

#### 5.4.1.35 HostAdapterName

`quickSpinStringNode` HostAdapterName

#### 5.4.1.36 HostAdapterVendor

`quickSpinStringNode` HostAdapterVendor

#### 5.4.1.37 IncompatibleDeviceCount

`quickSpinIntegerNode` IncompatibleDeviceCount

#### 5.4.1.38 IncompatibleDeviceID

`quickSpinStringNode` IncompatibleDeviceID

#### 5.4.1.39 IncompatibleDeviceModelName

`quickSpinStringNode` IncompatibleDeviceModelName

#### 5.4.1.40 IncompatibleDeviceSelector

`quickSpinIntegerNode` IncompatibleDeviceSelector

#### 5.4.1.41 IncompatibleDeviceVendorName

`quickSpinStringNode` IncompatibleDeviceVendorName

#### 5.4.1.42 IncompatibleGevDeviceIPAddress

`quickSpinIntegerNode` IncompatibleGevDeviceIPAddress



#### 5.4.1.43 IncompatibleGevDeviceMACAddress

`quickSpinIntegerNode` IncompatibleGevDeviceMACAddress

#### 5.4.1.44 IncompatibleGevDeviceSubnetMask

`quickSpinIntegerNode` IncompatibleGevDeviceSubnetMask

#### 5.4.1.45 InterfaceDisplayName

`quickSpinStringNode` InterfaceDisplayName

#### 5.4.1.46 InterfaceID

`quickSpinStringNode` InterfaceID

#### 5.4.1.47 InterfaceType

`quickSpinEnumerationNode` InterfaceType

#### 5.4.1.48 POEStatus

`quickSpinEnumerationNode` POEStatus

The documentation for this struct was generated from the following file:

- `include/spinc/TransportLayerInterfaceC.h`

## 5.5 `_quickSpinTLStream` Struct Reference

### Data Fields

- `quickSpinStringNode` `StreamID`
- `quickSpinEnumerationNode` `StreamType`
- `quickSpinIntegerNode` `StreamBufferCountManual`
- `quickSpinIntegerNode` `StreamBufferCountResult`
- `quickSpinIntegerNode` `StreamBufferCountMax`
- `quickSpinEnumerationNode` `StreamBufferCountMode`
- `quickSpinEnumerationNode` `StreamBufferHandlingMode`
- `quickSpinIntegerNode` `StreamAnnounceBufferMinimum`
- `quickSpinIntegerNode` `StreamAnnouncedBufferCount`
- `quickSpinIntegerNode` `StreamStartedFrameCount`
- `quickSpinIntegerNode` `StreamDeliveredFrameCount`
- `quickSpinIntegerNode` `StreamLostFrameCount`
- `quickSpinIntegerNode` `StreamInputBufferCount`
- `quickSpinIntegerNode` `StreamOutputBufferCount`
- `quickSpinBooleanNode` `StreamCRCCheckEnable`
- `quickSpinBooleanNode` `GevPacketResendMode`
- `quickSpinIntegerNode` `GevMaximumNumberResendRequests`
- `quickSpinIntegerNode` `GevPacketResendTimeout`
- `quickSpinBooleanNode` `StreamIsGrabbing`
- `quickSpinIntegerNode` `StreamChunkCountMaximum`
- `quickSpinIntegerNode` `StreamBufferAlignment`
- `quickSpinIntegerNode` `GevTotalPacketCount`
- `quickSpinIntegerNode` `GevFailedPacketCount`
- `quickSpinIntegerNode` `GevResendPacketCount`
- `quickSpinIntegerNode` `StreamFailedBufferCount`
- `quickSpinIntegerNode` `GevResendRequestCount`
- `quickSpinIntegerNode` `StreamBlockTransferSize`

### 5.5.1 Field Documentation

#### 5.5.1.1 `GevFailedPacketCount`

`quickSpinIntegerNode` `GevFailedPacketCount`

#### 5.5.1.2 `GevMaximumNumberResendRequests`

`quickSpinIntegerNode` `GevMaximumNumberResendRequests`

#### 5.5.1.3 GevPacketResendMode

[quickSpinBooleanNode](#) GevPacketResendMode

#### 5.5.1.4 GevPacketResendTimeout

[quickSpinIntegerNode](#) GevPacketResendTimeout

#### 5.5.1.5 GevResendPacketCount

[quickSpinIntegerNode](#) GevResendPacketCount

#### 5.5.1.6 GevResendRequestCount

[quickSpinIntegerNode](#) GevResendRequestCount

#### 5.5.1.7 GevTotalPacketCount

[quickSpinIntegerNode](#) GevTotalPacketCount

#### 5.5.1.8 StreamAnnounceBufferMinimum

[quickSpinIntegerNode](#) StreamAnnounceBufferMinimum

#### 5.5.1.9 StreamAnnouncedBufferCount

[quickSpinIntegerNode](#) StreamAnnouncedBufferCount

#### 5.5.1.10 StreamBlockTransferSize

[quickSpinIntegerNode](#) StreamBlockTransferSize

#### 5.5.1.11 StreamBufferAlignment

`quickSpinIntegerNode` StreamBufferAlignment

#### 5.5.1.12 StreamBufferCountManual

`quickSpinIntegerNode` StreamBufferCountManual

#### 5.5.1.13 StreamBufferCountMax

`quickSpinIntegerNode` StreamBufferCountMax

#### 5.5.1.14 StreamBufferCountMode

`quickSpinEnumerationNode` StreamBufferCountMode

#### 5.5.1.15 StreamBufferCountResult

`quickSpinIntegerNode` StreamBufferCountResult

#### 5.5.1.16 StreamBufferHandlingMode

`quickSpinEnumerationNode` StreamBufferHandlingMode

#### 5.5.1.17 StreamChunkCountMaximum

`quickSpinIntegerNode` StreamChunkCountMaximum

#### 5.5.1.18 StreamCRCCheckEnable

`quickSpinBooleanNode` StreamCRCCheckEnable

#### 5.5.1.19 StreamDeliveredFrameCount

`quickSpinIntegerNode` StreamDeliveredFrameCount

#### 5.5.1.20 StreamFailedBufferCount

`quickSpinIntegerNode` StreamFailedBufferCount

#### 5.5.1.21 StreamID

`quickSpinStringNode` StreamID

#### 5.5.1.22 StreamInputBufferCount

`quickSpinIntegerNode` StreamInputBufferCount

#### 5.5.1.23 StreamIsGrabbing

`quickSpinBooleanNode` StreamIsGrabbing

#### 5.5.1.24 StreamLostFrameCount

`quickSpinIntegerNode` StreamLostFrameCount

#### 5.5.1.25 StreamOutputBufferCount

`quickSpinIntegerNode` StreamOutputBufferCount

#### 5.5.1.26 StreamStartedFrameCount

`quickSpinIntegerNode` StreamStartedFrameCount

### 5.5.1.27 StreamType

[quickSpinEnumerationNode](#) StreamType

The documentation for this struct was generated from the following file:

- [include/spinc/TransportLayerStreamC.h](#)

## 5.6 \_quickSpinTLSystem Struct Reference

### Data Fields

- [quickSpinBooleanNode](#) EnumerateGEVInterfaces
- [quickSpinStringNode](#) TLID
- [quickSpinStringNode](#) TLVendorName
- [quickSpinStringNode](#) TLModelName
- [quickSpinStringNode](#) TLVersion
- [quickSpinStringNode](#) TLFileName
- [quickSpinStringNode](#) TLDisplayName
- [quickSpinStringNode](#) TLPath
- [quickSpinEnumerationNode](#) TLType
- [quickSpinIntegerNode](#) GenTLVersionMajor
- [quickSpinIntegerNode](#) GenTLVersionMinor
- [quickSpinIntegerNode](#) GenTLSFNCVersionMajor
- [quickSpinIntegerNode](#) GenTLSFNCVersionMinor
- [quickSpinIntegerNode](#) GenTLSFNCVersionSubMinor
- [quickSpinIntegerNode](#) GevVersionMajor
- [quickSpinIntegerNode](#) GevVersionMinor
- [quickSpinCommandNode](#) InterfaceUpdateList
- [quickSpinIntegerNode](#) InterfaceSelector
- [quickSpinStringNode](#) InterfaceID
- [quickSpinStringNode](#) InterfaceDisplayName
- [quickSpinIntegerNode](#) GevInterfaceMACAddress
- [quickSpinIntegerNode](#) GevInterfaceDefaultIPAddress
- [quickSpinIntegerNode](#) GevInterfaceDefaultSubnetMask
- [quickSpinIntegerNode](#) GevInterfaceDefaultGateway

### 5.6.1 Field Documentation

#### 5.6.1.1 EnumerateGEVInterfaces

[quickSpinBooleanNode](#) EnumerateGEVInterfaces

### 5.6.1.2 GenTLFNCVersionMajor

`quickSpinIntegerNode` GenTLFNCVersionMajor

### 5.6.1.3 GenTLFNCVersionMinor

`quickSpinIntegerNode` GenTLFNCVersionMinor

### 5.6.1.4 GenTLFNCVersionSubMinor

`quickSpinIntegerNode` GenTLFNCVersionSubMinor

### 5.6.1.5 GenTLVersionMajor

`quickSpinIntegerNode` GenTLVersionMajor

### 5.6.1.6 GenTLVersionMinor

`quickSpinIntegerNode` GenTLVersionMinor

### 5.6.1.7 GevInterfaceDefaultGateway

`quickSpinIntegerNode` GevInterfaceDefaultGateway

### 5.6.1.8 GevInterfaceDefaultIPAddress

`quickSpinIntegerNode` GevInterfaceDefaultIPAddress

### 5.6.1.9 GevInterfaceDefaultSubnetMask

`quickSpinIntegerNode` GevInterfaceDefaultSubnetMask

#### 5.6.1.10 **GevInterfaceMACAddress**

`quickSpinIntegerNode` `GevInterfaceMACAddress`

#### 5.6.1.11 **GevVersionMajor**

`quickSpinIntegerNode` `GevVersionMajor`

#### 5.6.1.12 **GevVersionMinor**

`quickSpinIntegerNode` `GevVersionMinor`

#### 5.6.1.13 **InterfaceDisplayName**

`quickSpinStringNode` `InterfaceDisplayName`

#### 5.6.1.14 **InterfaceID**

`quickSpinStringNode` `InterfaceID`

#### 5.6.1.15 **InterfaceSelector**

`quickSpinIntegerNode` `InterfaceSelector`

#### 5.6.1.16 **InterfaceUpdateList**

`quickSpinCommandNode` `InterfaceUpdateList`

#### 5.6.1.17 **TLDisplayName**

`quickSpinStringNode` `TLDisplayName`



#### 5.6.1.18 TLFileName

`quickSpinStringNode` TLFileName

#### 5.6.1.19 TLID

`quickSpinStringNode` TLID

#### 5.6.1.20 TLModelName

`quickSpinStringNode` TLModelName

#### 5.6.1.21 TLPath

`quickSpinStringNode` TLPath

#### 5.6.1.22 TLType

`quickSpinEnumerationNode` TLType

#### 5.6.1.23 TLVendorName

`quickSpinStringNode` TLVendorName

#### 5.6.1.24 TLVersion

`quickSpinStringNode` TLVersion

The documentation for this struct was generated from the following file:

- `include/spinc/TransportLayerSystemC.h`

## 5.7 `_spinAVIOption` Struct Reference

Options for saving uncompressed videos.

### Data Fields

- float `frameRate`  
*Frame rate of the stream.*
- unsigned int `reserved` [256]  
*Reserved for future use.*

### 5.7.1 Detailed Description

Options for saving uncompressed videos.

Used in saving AVI videos with a call to `spinAVIRecorderOpenUncompressed()`.

### 5.7.2 Field Documentation

#### 5.7.2.1 `frameRate`

```
float frameRate
```

Frame rate of the stream.

#### 5.7.2.2 `reserved`

```
unsigned int reserved[256]
```

Reserved for future use.

The documentation for this struct was generated from the following file:

- `include/spinc/SpinnakerDefsC.h`

## 5.8 `_spinBMPOption` Struct Reference

Options for saving BMP images.

## Data Fields

- [bool8\\_t indexedColor\\_8bit](#)
- unsigned int [reserved](#) [16]

*Reserved for future use.*

### 5.8.1 Detailed Description

Options for saving BMP images.

Used in saving PPM images with a call to [spinImageSaveBmp\(\)](#).

### 5.8.2 Field Documentation

#### 5.8.2.1 indexedColor\_8bit

[bool8\\_t](#) indexedColor\_8bit

#### 5.8.2.2 reserved

unsigned int reserved[16]

Reserved for future use.

The documentation for this struct was generated from the following file:

- include/spinc/[SpinnakerDefsC.h](#)

## 5.9 \_spinChunkData Struct Reference

The type of information that can be obtained from image chunk data.

## Data Fields

- double [m\\_blackLevel](#)
- int64\_t [m\\_frameID](#)
- double [m\\_exposureTime](#)
- int64\_t [m\\_timestamp](#)
- int64\_t [m\\_exposureEndLineStatusAll](#)
- int64\_t [m\\_width](#)
- int64\_t [m\\_image](#)
- int64\_t [m\\_height](#)
- double [m\\_gain](#)
- int64\_t [m\\_sequencerSetActive](#)
- int64\_t [m\\_cRC](#)
- int64\_t [m\\_offsetX](#)
- int64\_t [m\\_offsetY](#)
- int64\_t [m\\_serialDataLength](#)
- int64\_t [m\\_partSelector](#)
- int64\_t [m\\_pixelDynamicRangeMin](#)
- int64\_t [m\\_pixelDynamicRangeMax](#)
- int64\_t [m\\_timestampLatchValue](#)
- int64\_t [m\\_lineStatusAll](#)
- int64\_t [m\\_counterValue](#)
- double [m\\_timerValue](#)
- int64\_t [m\\_scanLineSelector](#)
- int64\_t [m\\_encoderValue](#)
- int64\_t [m\\_linePitch](#)
- int64\_t [m\\_transferBlockID](#)
- int64\_t [m\\_transferQueueCurrentBlockCount](#)
- int64\_t [m\\_streamChannelID](#)
- double [m\\_scan3dCoordinateScale](#)
- double [m\\_scan3dCoordinateOffset](#)
- double [m\\_scan3dInvalidDataValue](#)
- double [m\\_scan3dAxisMin](#)
- double [m\\_scan3dAxisMax](#)
- double [m\\_scan3dTransformValue](#)
- double [m\\_scan3dCoordinateReferenceValue](#)
- int64\_t [m\\_inferenceFrameId](#)
- int64\_t [m\\_inferenceResult](#)
- double [m\\_inferenceConfidence](#)

### 5.9.1 Detailed Description

The type of information that can be obtained from image chunk data.

### 5.9.2 Field Documentation

#### 5.9.2.1 [m\\_blackLevel](#)

```
double m_blackLevel
```

#### 5.9.2.2 m\_counterValue

```
int64_t m_counterValue
```

#### 5.9.2.3 m\_cRC

```
int64_t m_cRC
```

#### 5.9.2.4 m\_encoderValue

```
int64_t m_encoderValue
```

#### 5.9.2.5 m\_exposureEndLineStatusAll

```
int64_t m_exposureEndLineStatusAll
```

#### 5.9.2.6 m\_exposureTime

```
double m_exposureTime
```

#### 5.9.2.7 m\_frameID

```
int64_t m_frameID
```

#### 5.9.2.8 m\_gain

```
double m_gain
```

#### 5.9.2.9 m\_height

```
int64_t m_height
```

**5.9.2.10 m\_image**

```
int64_t m_image
```

**5.9.2.11 m\_inferenceConfidence**

```
double m_inferenceConfidence
```

**5.9.2.12 m\_inferenceFrameId**

```
int64_t m_inferenceFrameId
```

**5.9.2.13 m\_inferenceResult**

```
int64_t m_inferenceResult
```

**5.9.2.14 m\_linePitch**

```
int64_t m_linePitch
```

**5.9.2.15 m\_lineStatusAll**

```
int64_t m_lineStatusAll
```

**5.9.2.16 m\_offsetX**

```
int64_t m_offsetX
```

**5.9.2.17 m\_offsetY**

```
int64_t m_offsetY
```

**5.9.2.18 m\_partSelector**

```
int64_t m_partSelector
```

**5.9.2.19 m\_pixelDynamicRangeMax**

```
int64_t m_pixelDynamicRangeMax
```

**5.9.2.20 m\_pixelDynamicRangeMin**

```
int64_t m_pixelDynamicRangeMin
```

**5.9.2.21 m\_scan3dAxisMax**

```
double m_scan3dAxisMax
```

**5.9.2.22 m\_scan3dAxisMin**

```
double m_scan3dAxisMin
```

**5.9.2.23 m\_scan3dCoordinateOffset**

```
double m_scan3dCoordinateOffset
```

**5.9.2.24 m\_scan3dCoordinateReferenceValue**

```
double m_scan3dCoordinateReferenceValue
```

**5.9.2.25 m\_scan3dCoordinateScale**

```
double m_scan3dCoordinateScale
```

**5.9.2.26 m\_scan3dInvalidDataValue**

```
double m_scan3dInvalidDataValue
```

**5.9.2.27 m\_scan3dTransformValue**

```
double m_scan3dTransformValue
```

**5.9.2.28 m\_scanLineSelector**

```
int64_t m_scanLineSelector
```

**5.9.2.29 m\_sequencerSetActive**

```
int64_t m_sequencerSetActive
```

**5.9.2.30 m\_serialDataLength**

```
int64_t m_serialDataLength
```

**5.9.2.31 m\_streamChannelID**

```
int64_t m_streamChannelID
```

**5.9.2.32 m\_timerValue**

```
double m_timerValue
```

**5.9.2.33 m\_timestamp**

```
int64_t m_timestamp
```



#### 5.9.2.34 m\_timestampLatchValue

```
int64_t m_timestampLatchValue
```

#### 5.9.2.35 m\_transferBlockID

```
int64_t m_transferBlockID
```

#### 5.9.2.36 m\_transferQueueCurrentBlockCount

```
int64_t m_transferQueueCurrentBlockCount
```

#### 5.9.2.37 m\_width

```
int64_t m_width
```

The documentation for this struct was generated from the following file:

- include/spinc/[ChunkDataDefC.h](#)

## 5.10 \_spinH264Option Struct Reference

Options for saving H264 videos.

### Data Fields

- float [frameRate](#)  
*Frame rate of the stream.*
- unsigned int [width](#)  
*Width of source image.*
- unsigned int [height](#)  
*Height of source image.*
- unsigned int [bitrate](#)  
*Bitrate to encode at.*
- unsigned int [reserved](#) [256]  
*Reserved for future use.*

### 5.10.1 Detailed Description

Options for saving H264 videos.

Used in saving H264 videos with a call to `spinAVIRecorderOpenH264()`.

### 5.10.2 Field Documentation

#### 5.10.2.1 bitrate

```
unsigned int bitrate
```

Bitrate to encode at.

#### 5.10.2.2 frameRate

```
float frameRate
```

Frame rate of the stream.

#### 5.10.2.3 height

```
unsigned int height
```

Height of source image.

#### 5.10.2.4 reserved

```
unsigned int reserved[256]
```

Reserved for future use.

#### 5.10.2.5 width

```
unsigned int width
```

Width of source image.

The documentation for this struct was generated from the following file:

- `include/spinc/SpinnakerDefsC.h`

## 5.11 \_spinJPEGOption Struct Reference

Options for saving JPEG images.

### Data Fields

- `bool8_t progressive`  
*Whether to save as a progressive JPEG file.*
- unsigned int `quality`  
*JPEG image quality in range (0-100).*
- unsigned int `reserved` [16]  
*Reserved for future use.*

### 5.11.1 Detailed Description

Options for saving JPEG images.

Used in saving PPM images with a call to `spinImageSaveJpeg()`.

### 5.11.2 Field Documentation

#### 5.11.2.1 progressive

`bool8_t progressive`

Whether to save as a progressive JPEG file.

#### 5.11.2.2 quality

`unsigned int quality`

JPEG image quality in range (0-100).

- 100 - Superb quality.
- 75 - Good quality.
- 50 - Normal quality.
- 10 - Poor quality.

### 5.11.2.3 reserved

```
unsigned int reserved[16]
```

Reserved for future use.

The documentation for this struct was generated from the following file:

- include/spinc/[SpinnakerDefsC.h](#)

## 5.12 \_spinJPG2Option Struct Reference

Options for saving JPEG 2000 images.

### Data Fields

- unsigned int [quality](#)  
*JPEG saving quality in range (1-512).*
- unsigned int [reserved](#) [16]  
*Reserved for future use.*

### 5.12.1 Detailed Description

Options for saving JPEG 2000 images.

Used in saving PPM images with a call to [spinImageSaveJpg2\(\)](#).

### 5.12.2 Field Documentation

#### 5.12.2.1 quality

```
unsigned int quality
```

JPEG saving quality in range (1-512).

#### 5.12.2.2 reserved

```
unsigned int reserved[16]
```

Reserved for future use.

The documentation for this struct was generated from the following file:

- include/spinc/[SpinnakerDefsC.h](#)

## 5.13 \_spinLibraryVersion Struct Reference

Provides easier access to the current version of Spinnaker.

### Data Fields

- unsigned int [major](#)  
*Major version of the library.*
- unsigned int [minor](#)  
*Minor version of the library.*
- unsigned int [type](#)  
*Version type of the library.*
- unsigned int [build](#)  
*Build number of the library.*

### 5.13.1 Detailed Description

Provides easier access to the current version of Spinnaker.

### 5.13.2 Field Documentation

#### 5.13.2.1 build

```
unsigned int build
```

Build number of the library.

#### 5.13.2.2 major

```
unsigned int major
```

Major version of the library.

#### 5.13.2.3 minor

```
unsigned int minor
```

Minor version of the library.

#### 5.13.2.4 type

`unsigned int type`

Version type of the library.

The documentation for this struct was generated from the following file:

- `include/spinc/SpinnakerDefsC.h`

### 5.14 \_\_spinMJPGOption Struct Reference

Options for saving MJPG videos.

#### Data Fields

- float `frameRate`  
*Frame rate of the stream.*
- unsigned int `quality`  
*Image quality (1-100)*
- unsigned int `reserved` [256]

#### 5.14.1 Detailed Description

Options for saving MJPG videos.

Used in saving MJPG videos with a call to `spinAVIRecorderOpenMJPG()`.

#### 5.14.2 Field Documentation

##### 5.14.2.1 frameRate

`float frameRate`

Frame rate of the stream.

##### 5.14.2.2 quality

`unsigned int quality`

Image quality (1-100)

### 5.14.2.3 reserved

```
unsigned int reserved[256]
```

The documentation for this struct was generated from the following file:

- include/spinc/[SpinnakerDefsC.h](#)

## 5.15 \_spinPGMOption Struct Reference

Options for saving PGM images.

### Data Fields

- [bool8\\_t](#) `binaryFile`  
*Whether to save the PPM as a binary file.*
- unsigned int [reserved](#) [16]  
*Reserved for future use.*

### 5.15.1 Detailed Description

Options for saving PGM images.

### 5.15.2 Field Documentation

#### 5.15.2.1 binaryFile

```
bool8_t binaryFile
```

Whether to save the PPM as a binary file.

#### 5.15.2.2 reserved

```
unsigned int reserved[16]
```

Reserved for future use.

The documentation for this struct was generated from the following file:

- include/spinc/[SpinnakerDefsC.h](#)

## 5.16 \_spinPNGOption Struct Reference

Options for saving PNG images.

### Data Fields

- [bool8\\_t interlaced](#)  
*Whether to save the PNG as interlaced.*
- unsigned int [compressionLevel](#)  
*Compression level (0-9).*
- unsigned int [reserved](#) [16]  
*Reserved for future use.*

### 5.16.1 Detailed Description

Options for saving PNG images.

Used in saving PNG images with a call to [spinImageSavePng\(\)](#).

### 5.16.2 Field Documentation

#### 5.16.2.1 compressionLevel

```
unsigned int compressionLevel
```

Compression level (0-9).

0 is no compression, 9 is best compression.

#### 5.16.2.2 interlaced

```
bool8_t interlaced
```

Whether to save the PNG as interlaced.

#### 5.16.2.3 reserved

```
unsigned int reserved[16]
```

Reserved for future use.

The documentation for this struct was generated from the following file:

- include/spinc/[SpinnakerDefsC.h](#)



## 5.17 \_spinPPMOption Struct Reference

Options for saving PPM images.

### Data Fields

- `bool8_t binaryFile`  
*Whether to save the PPM as a binary file.*
- unsigned int `reserved` [16]  
*Reserved for future use.*

### 5.17.1 Detailed Description

Options for saving PPM images.

Used in saving PPM images with a call to `spinImageSavePpm()`.

### 5.17.2 Field Documentation

#### 5.17.2.1 binaryFile

`bool8_t binaryFile`

Whether to save the PPM as a binary file.

#### 5.17.2.2 reserved

`unsigned int reserved[16]`

Reserved for future use.

The documentation for this struct was generated from the following file:

- `include/spinc/SpinnakerDefsC.h`

## 5.18 \_spinTIFFOption Struct Reference

Options for saving TIFF images.

## Data Fields

- spinCompressionMethod [compression](#)  
*Compression method to use for encoding TIFF images.*
- unsigned int [reserved](#) [16]  
*Reserved for future use.*

### 5.18.1 Detailed Description

Options for saving TIFF images.

Used in saving PPM images with a call to [spinImageSaveTiff\(\)](#).

### 5.18.2 Field Documentation

#### 5.18.2.1 compression

```
spinCompressionMethod compression
```

Compression method to use for encoding TIFF images.

#### 5.18.2.2 reserved

```
unsigned int reserved[16]
```

Reserved for future use.

The documentation for this struct was generated from the following file:

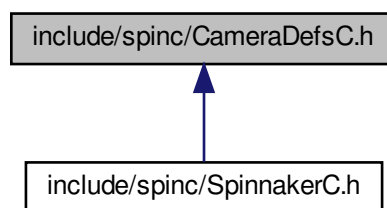
- include/spinc/[SpinnakerDefsC.h](#)

## Chapter 6

# File Documentation

### 6.1 include/spinc/CameraDefsC.h File Reference

This graph shows which files directly or indirectly include this file:



### Enumerations

- `enum _spinLUTSelectorEnums {  
    LUTSelector_LUT1,  
    NUM_LUTSELECTOR }`  
*The enum definitions for camera nodes.*
- `enum _spinExposureModeEnums {  
    ExposureMode_Timed,  
    ExposureMode_TriggerWidth,  
    NUM_EXPOSUREMODE }`
- `enum _spinAcquisitionModeEnums {  
    AcquisitionMode_Continuous,  
    AcquisitionMode_SingleFrame,  
    AcquisitionMode_MultiFrame,  
    NUM_ACQUISITIONMODE }`

- `enum _spinTriggerSourceEnums {`  
    `TriggerSource_Software,`  
    `TriggerSource_Line0,`  
    `TriggerSource_Line1,`  
    `TriggerSource_Line2,`  
    `TriggerSource_Line3,`  
    `TriggerSource_UserOutput0,`  
    `TriggerSource_UserOutput1,`  
    `TriggerSource_UserOutput2,`  
    `TriggerSource_UserOutput3,`  
    `TriggerSource_Counter0Start,`  
    `TriggerSource_Counter1Start,`  
    `TriggerSource_Counter0End,`  
    `TriggerSource_Counter1End,`  
    `TriggerSource_LogicBlock0,`  
    `TriggerSource_LogicBlock1,`  
    `TriggerSource_Action0,`  
    `NUM_TRIGGERSOURCE }`
- `enum _spinTriggerActivationEnums {`  
    `TriggerActivation_LevelLow,`  
    `TriggerActivation_LevelHigh,`  
    `TriggerActivation_FallingEdge,`  
    `TriggerActivation_RisingEdge,`  
    `TriggerActivation_AnyEdge,`  
    `NUM_TRIGGERACTIVATION }`
- `enum _spinSensorShutterModeEnums {`  
    `SensorShutterMode_Global,`  
    `SensorShutterMode_Rolling,`  
    `SensorShutterMode_GlobalReset,`  
    `NUM_SENSORSHUTTERMODE }`
- `enum _spinTriggerModeEnums {`  
    `TriggerMode_Off,`  
    `TriggerMode_On,`  
    `NUM_TRIGGERMODE }`
- `enum _spinTriggerOverlapEnums {`  
    `TriggerOverlap_Off,`  
    `TriggerOverlap_ReadOut,`  
    `TriggerOverlap_PreviousFrame,`  
    `NUM_TRIGGEROVERLAP }`
- `enum _spinTriggerSelectorEnums {`  
    `TriggerSelector_AcquisitionStart,`  
    `TriggerSelector_FrameStart,`  
    `TriggerSelector_FrameBurstStart,`  
    `NUM_TRIGGERSELECTOR }`
- `enum _spinExposureAutoEnums {`  
    `ExposureAuto_Off,`  
    `ExposureAuto_Once,`  
    `ExposureAuto_Continuous,`  
    `NUM_EXPOSUREAUTO }`
- `enum _spinEventSelectorEnums {`  
    `EventSelector_Error,`  
    `EventSelector_ExposureEnd,`  
    `EventSelector_SerialPortReceive,`  
    `NUM_EVENTSELECTOR }`
- `enum _spinEventNotificationEnums {`  
    `EventNotification_On,`  
    `EventNotification_Off,`  
    `NUM_EVENTNOTIFICATION }`

- enum `_spinLogicBlockSelectorEnums` {  
`LogicBlockSelector_LogicBlock0,`  
`LogicBlockSelector_LogicBlock1,`  
`NUM_LOGICBLOCKSELECTOR` }
- enum `_spinLogicBlockLUTInputActivationEnums` {  
`LogicBlockLUTInputActivation_LevelLow,`  
`LogicBlockLUTInputActivation_LevelHigh,`  
`LogicBlockLUTInputActivation_FallingEdge,`  
`LogicBlockLUTInputActivation_RisingEdge,`  
`LogicBlockLUTInputActivation_AnyEdge,`  
`NUM_LOGICBLOCKLUTINPUTACTIVATION` }
- enum `_spinLogicBlockLUTInputSelectorEnums` {  
`LogicBlockLUTInputSelector_Input0,`  
`LogicBlockLUTInputSelector_Input1,`  
`LogicBlockLUTInputSelector_Input2,`  
`LogicBlockLUTInputSelector_Input3,`  
`NUM_LOGICBLOCKLUTINPUTSELECTOR` }
- enum `_spinLogicBlockLUTInputSourceEnums` {  
`LogicBlockLUTInputSource_Zero,`  
`LogicBlockLUTInputSource_Line0,`  
`LogicBlockLUTInputSource_Line1,`  
`LogicBlockLUTInputSource_Line2,`  
`LogicBlockLUTInputSource_Line3,`  
`LogicBlockLUTInputSource_UserOutput0,`  
`LogicBlockLUTInputSource_UserOutput1,`  
`LogicBlockLUTInputSource_UserOutput2,`  
`LogicBlockLUTInputSource_UserOutput3,`  
`LogicBlockLUTInputSource_Counter0Start,`  
`LogicBlockLUTInputSource_Counter1Start,`  
`LogicBlockLUTInputSource_Counter0End,`  
`LogicBlockLUTInputSource_Counter1End,`  
`LogicBlockLUTInputSource_LogicBlock0,`  
`LogicBlockLUTInputSource_LogicBlock1,`  
`LogicBlockLUTInputSource_ExposureStart,`  
`LogicBlockLUTInputSource_ExposureEnd,`  
`LogicBlockLUTInputSource_FrameTriggerWait,`  
`LogicBlockLUTInputSource_AcquisitionActive,`  
`NUM_LOGICBLOCKLUTINPUTSOURCE` }
- enum `_spinLogicBlockLUTSelectorEnums` {  
`LogicBlockLUTSelector_Value,`  
`LogicBlockLUTSelector_Enable,`  
`NUM_LOGICBLOCKLUTSELECTOR` }
- enum `_spinColorTransformationSelectorEnums` {  
`ColorTransformationSelector_RGBtoRGB,`  
`ColorTransformationSelector_RGBtoYUV,`  
`NUM_COLORTRANSFORMATIONSELECTOR` }
- enum `_spinRgbTransformLightSourceEnums` {  
`RgbTransformLightSource_General,`  
`RgbTransformLightSource_Tungsten2800K,`  
`RgbTransformLightSource_WarmFluorescent3000K,`  
`RgbTransformLightSource_CoolFluorescent4000K,`  
`RgbTransformLightSource_Daylight5000K,`  
`RgbTransformLightSource_Cloudy6500K,`  
`RgbTransformLightSource_Shade8000K,`  
`RgbTransformLightSource_Custom,`  
`NUM_RGBTRANSFORMLIGHTSOURCE` }
- enum `_spinColorTransformationValueSelectorEnums` {  
`ColorTransformationValueSelector_Gain00,`

```

ColorTransformationValueSelector_Gain01,
ColorTransformationValueSelector_Gain02,
ColorTransformationValueSelector_Gain10,
ColorTransformationValueSelector_Gain11,
ColorTransformationValueSelector_Gain12,
ColorTransformationValueSelector_Gain20,
ColorTransformationValueSelector_Gain21,
ColorTransformationValueSelector_Gain22,
ColorTransformationValueSelector_Offset0,
ColorTransformationValueSelector_Offset1,
ColorTransformationValueSelector_Offset2,
NUM_COLORTRANSFORMATIONVALUESELECTOR }

• enum _spinDeviceRegistersEndiannessEnums {
    DeviceRegistersEndianness_Little,
    DeviceRegistersEndianness_Big,
    NUM_DEVICEREGISTERSENDIANNES }

• enum _spinDeviceScanTypeEnums {
    DeviceScanType_Areascan,
    NUM_DEVICESCANTYPE }

• enum _spinDeviceCharacterSetEnums {
    DeviceCharacterSet_UTF8,
    DeviceCharacterSet_ASCII,
    NUM_DEVICECHARACTERSET }

• enum _spinDeviceTLTypeEnums {
    DeviceTLType_GigEVision,
    DeviceTLType_CameraLink,
    DeviceTLType_CameraLinkHS,
    DeviceTLType_CoaXPRESS,
    DeviceTLType_USB3Vision,
    DeviceTLType_Custom,
    NUM_DEVICETLTYPE }

• enum _spinDevicePowerSupplySelectorEnums {
    DevicePowerSupplySelector_External,
    NUM_DEVICEPOWERSUPPLYSELECTOR }

• enum _spinDeviceTemperatureSelectorEnums {
    DeviceTemperatureSelector_Sensor,
    NUM_DEVICETEMPERATURESELECTOR }

• enum _spinDeviceIndicatorModeEnums {
    DeviceIndicatorMode_Inactive,
    DeviceIndicatorMode_Active,
    DeviceIndicatorMode_ErrorStatus,
    NUM_DEVICEINDICATORMODE }

• enum _spinAutoExposureControlPriorityEnums {
    AutoExposureControlPriority_Gain,
    AutoExposureControlPriority_ExposureTime,
    NUM_AUTOEXPOSURECONTROLPRIORITY }

• enum _spinAutoExposureMeteringModeEnums {
    AutoExposureMeteringMode_Average,
    AutoExposureMeteringMode_Spot,
    AutoExposureMeteringMode_Partial,
    AutoExposureMeteringMode_CenterWeighted,
    AutoExposureMeteringMode_HistogramPeak,
    NUM_AUTOEXPOSUREMETERINGMODE }

• enum _spinBalanceWhiteAutoProfileEnums {
    BalanceWhiteAutoProfile_Indoor,
    BalanceWhiteAutoProfile_Outdoor,
    NUM_BALANCEWHITEAUTOPROFILE }

```

- enum `_spinAutoAlgorithmSelectorEnums` {  
`AutoAlgorithmSelector_Awb,`  
`AutoAlgorithmSelector_Ae,`  
`NUM_AUTOALGORITHMSELECTOR` }
- enum `_spinAutoExposureTargetGreyValueAutoEnums` {  
`AutoExposureTargetGreyValueAuto_Off,`  
`AutoExposureTargetGreyValueAuto_Continuous,`  
`NUM_AUTOEXPOSURETARGETGREYVALUEAUTO` }
- enum `_spinAutoExposureLightingModeEnums` {  
`AutoExposureLightingMode_AutoDetect,`  
`AutoExposureLightingMode_Backlight,`  
`AutoExposureLightingMode_Frontlight,`  
`AutoExposureLightingMode_Normal,`  
`NUM_AUTOEXPOSURELIGHTINGMODE` }
- enum `_spinGevIEEE1588StatusEnums` {  
`GevIEEE1588Status_Initializing,`  
`GevIEEE1588Status_Faulty,`  
`GevIEEE1588Status_Disabled,`  
`GevIEEE1588Status_Listening,`  
`GevIEEE1588Status_PreMaster,`  
`GevIEEE1588Status_Master,`  
`GevIEEE1588Status_Passive,`  
`GevIEEE1588Status_Uncalibrated,`  
`GevIEEE1588Status_Slave,`  
`NUM_GEVIEEE1588STATUS` }
- enum `_spinGevIEEE1588ModeEnums` {  
`GevIEEE1588Mode_Auto,`  
`GevIEEE1588Mode_SlaveOnly,`  
`NUM_GEVIEEE1588MODE` }
- enum `_spinGevIEEE1588ClockAccuracyEnums` {  
`GevIEEE1588ClockAccuracy_Unknown,`  
`NUM_GEVIEEE1588CLOCKACCURACY` }
- enum `_spinGevCCPEnums` {  
`GevCCP_OpenAccess,`  
`GevCCP_ExclusiveAccess,`  
`GevCCP_ControlAccess,`  
`NUM_GEVCCP` }
- enum `_spinGevSupportedOptionSelectorEnums` {  
`GevSupportedOptionSelector_UserDefinedName,`  
`GevSupportedOptionSelector_SerialNumber,`  
`GevSupportedOptionSelector_HeartbeatDisable,`  
`GevSupportedOptionSelector_LinkSpeed,`  
`GevSupportedOptionSelector_CCPApplicationSocket,`  
`GevSupportedOptionSelector_ManifestTable,`  
`GevSupportedOptionSelector_TestData,`  
`GevSupportedOptionSelector_DiscoveryAckDelay,`  
`GevSupportedOptionSelector_DiscoveryAckDelayWritable,`  
`GevSupportedOptionSelector_ExtendedStatusCodes,`  
`GevSupportedOptionSelector_Action,`  
`GevSupportedOptionSelector_PendingAck,`  
`GevSupportedOptionSelector_EventData,`  
`GevSupportedOptionSelector_Event,`  
`GevSupportedOptionSelector_PacketResend,`  
`GevSupportedOptionSelector_WriteMem,`  
`GevSupportedOptionSelector_CommandsConcatenation,`  
`GevSupportedOptionSelector_IPConfigurationLLA,`  
`GevSupportedOptionSelector_IPConfigurationDHCP,`  
`GevSupportedOptionSelector_IPConfigurationPersistentIP,`

```

GevSupportedOptionSelector_StreamChannelSourceSocket,
GevSupportedOptionSelector_MessageChannelSourceSocket,
NUM_GEVSUPPORTEDOPTIONSELECTOR }

• enum _spinBlackLevelSelectorEnums {
    BlackLevelSelector_All,
    BlackLevelSelector_Analog,
    BlackLevelSelector_Digital,
    NUM_BLACKLEVELSELECTOR }

• enum _spinBalanceWhiteAutoEnums {
    BalanceWhiteAuto_Off,
    BalanceWhiteAuto_Once,
    BalanceWhiteAuto_Continuous,
    NUM_BALANCEWHITEAUTO }

• enum _spinGainAutoEnums {
    GainAuto_Off,
    GainAuto_Once,
    GainAuto_Continuous,
    NUM_GAINAUTO }

• enum _spinBalanceRatioSelectorEnums {
    BalanceRatioSelector_Red,
    BalanceRatioSelector_Blue,
    NUM_BALANCERATIOSELECTOR }

• enum _spinGainSelectorEnums {
    GainSelector_All,
    NUM_GAINSELECTOR }

• enum _spinDefectCorrectionModeEnums {
    DefectCorrectionMode_Average,
    DefectCorrectionMode_Highlight,
    DefectCorrectionMode_Zero,
    NUM_DEFECTCORRECTIONMODE }

• enum _spinUserSetSelectorEnums {
    UserSetSelector_Default,
    UserSetSelector_UserSet0,
    UserSetSelector_UserSet1,
    NUM_USERSETSELECTOR }

• enum _spinUserSetDefaultEnums {
    UserSetDefault_Default,
    UserSetDefault_UserSet0,
    UserSetDefault_UserSet1,
    NUM_USERSETDEFAULT }

• enum _spinSerialPortBaudRateEnums {
    SerialPortBaudRate_Baud300,
    SerialPortBaudRate_Baud600,
    SerialPortBaudRate_Baud1200,
    SerialPortBaudRate_Baud2400,
    SerialPortBaudRate_Baud4800,
    SerialPortBaudRate_Baud9600,
    SerialPortBaudRate_Baud14400,
    SerialPortBaudRate_Baud19200,
    SerialPortBaudRate_Baud38400,
    SerialPortBaudRate_Baud57600,
    SerialPortBaudRate_Baud115200,
    SerialPortBaudRate_Baud230400,
    SerialPortBaudRate_Baud460800,
    SerialPortBaudRate_Baud921600,
    NUM_SERIALPORTBAUDRATE }

• enum _spinSerialPortParityEnums {
    SerialPortParity_None,

```



```
SerialPortParity_Odd,  
SerialPortParity_Even,  
SerialPortParity_Mark,  
SerialPortParity_Space,  
NUM_SERIALPORTPARITY }  
• enum _spinSerialPortSelectorEnums {  
    SerialPortSelector_SerialPort0,  
    NUM_SERIALPORTSELECTOR }  
• enum _spinSerialPortStopBitsEnums {  
    SerialPortStopBits_Bits1,  
    SerialPortStopBits_Bits1AndAHalf,  
    SerialPortStopBits_Bits2,  
    NUM_SERIALPORTSTOPBITS }  
• enum _spinSerialPortSourceEnums {  
    SerialPortSource_Line0,  
    SerialPortSource_Line1,  
    SerialPortSource_Line2,  
    SerialPortSource_Line3,  
    SerialPortSource_Off,  
    NUM_SERIALPORTSOURCE }  
• enum _spinSequencerModeEnums {  
    SequencerMode_Off,  
    SequencerMode_On,  
    NUM_SEQUENCERMODE }  
• enum _spinSequencerConfigurationValidEnums {  
    SequencerConfigurationValid_No,  
    SequencerConfigurationValid_Yes,  
    NUM_SEQUENCERCONFIGURATIONVALID }  
• enum _spinSequencerSetValidEnums {  
    SequencerSetValid_No,  
    SequencerSetValid_Yes,  
    NUM_SEQUENCERSETVALID }  
• enum _spinSequencerTriggerActivationEnums {  
    SequencerTriggerActivation_RisingEdge,  
    SequencerTriggerActivation_FallingEdge,  
    SequencerTriggerActivation_AnyEdge,  
    SequencerTriggerActivation_LevelHigh,  
    SequencerTriggerActivation_LevelLow,  
    NUM_SEQUENCERTRIGGERACTIVATION }  
• enum _spinSequencerConfigurationModeEnums {  
    SequencerConfigurationMode_Off,  
    SequencerConfigurationMode_On,  
    NUM_SEQUENCERCONFIGURATIONMODE }  
• enum _spinSequencerTriggerSourceEnums {  
    SequencerTriggerSource_Off,  
    SequencerTriggerSource_FrameStart,  
    NUM_SEQUENCERTRIGGERSOURCE }  
• enum _spinTransferQueueModeEnums {  
    TransferQueueMode_FirstInFirstOut,  
    NUM_TRANSFERQUEUEMODE }  
• enum _spinTransferOperationModeEnums {  
    TransferOperationMode_Continuous,  
    TransferOperationMode_MultiBlock,  
    NUM_TRANSFEROPERATIONMODE }  
• enum _spinTransferControlModeEnums {  
    TransferControlMode_Basic,  
    TransferControlMode_Automatic,
```

```

TransferControlMode_UserControlled,
NUM_TRANSFERCONTROLMODE }

• enum _spinChunkGainSelectorEnums {
    ChunkGainSelector_All,
    ChunkGainSelector_Red,
    ChunkGainSelector_Green,
    ChunkGainSelector_Blue,
    NUM_CHUNKGAINSELECTOR }

• enum _spinChunkSelectorEnums {
    ChunkSelector_Image,
    ChunkSelector_CRC,
    ChunkSelector_FrameID,
    ChunkSelector_OffsetX,
    ChunkSelector_OffsetY,
    ChunkSelector_Width,
    ChunkSelector_Height,
    ChunkSelector_ExposureTime,
    ChunkSelector_Gain,
    ChunkSelector_BlackLevel,
    ChunkSelector_PixelFormat,
    ChunkSelector_Timestamp,
    ChunkSelector_SequencerSetActive,
    ChunkSelector_SerialData,
    ChunkSelector_ExposureEndLineStatusAll,
    NUM_CHUNKSELECTOR }

• enum _spinChunkBlackLevelSelectorEnums {
    ChunkBlackLevelSelector_All,
    NUM_CHUNKBLACKLEVELSELECTOR }

• enum _spinChunkPixelFormatEnums {
    ChunkPixelFormat_Mono8,
    ChunkPixelFormat_Mono12Packed,
    ChunkPixelFormat_Mono16,
    ChunkPixelFormat_RGB8Packed,
    ChunkPixelFormat_YUV422Packed,
    ChunkPixelFormat_BayerGR8,
    ChunkPixelFormat_BayerRG8,
    ChunkPixelFormat_BayerGB8,
    ChunkPixelFormat_BayerBG8,
    ChunkPixelFormat_YCbCr601_422_8_CbYCrY,
    NUM_CHUNKPIXELFORMAT }

• enum _spinFileOperationStatusEnums {
    FileOperationStatus_Success,
    FileOperationStatus_Failure,
    FileOperationStatus_Overflow,
    NUM_FILEOPERATIONSTATUS }

• enum _spinFileOpenModeEnums {
    FileOpenMode_Read,
    FileOpenMode_Write,
    FileOpenMode_ReadWrite,
    NUM_FILEOPENMODE }

• enum _spinFileOperationSelectorEnums {
    FileOperationSelector_Open,
    FileOperationSelector_Close,
    FileOperationSelector_Read,
    FileOperationSelector_Write,
    FileOperationSelector_Delete,
    NUM_FILEOPERATIONSELECTOR }

```

- enum `_spinFileSelectorEnums` {  
    FileSelector\_UserSetDefault,  
    FileSelector\_UserSet0,  
    FileSelector\_UserSet1,  
    FileSelector\_UserFile1,  
    FileSelector\_SerialPort0,  
    NUM\_FILESELECTOR }
- enum `_spinBinningSelectorEnums` {  
    BinningSelector\_All,  
    BinningSelector\_Sensor,  
    BinningSelector\_ISP,  
    NUM\_BINNINGSELECTOR }
- enum `_spinTestPatternGeneratorSelectorEnums` {  
    TestPatternGeneratorSelector\_Sensor,  
    TestPatternGeneratorSelector\_PipelineStart,  
    NUM\_TESTPATTERNGENERATORSELECTOR }
- enum `_spinTestPatternEnums` {  
    TestPattern\_Off,  
    TestPattern\_Increment,  
    TestPattern\_SensorTestPattern,  
    NUM\_TESTPATTERN }
- enum `_spinPixelColorFilterEnums` {  
    PixelColorFilter\_None,  
    PixelColorFilter\_BayerRG,  
    PixelColorFilter\_BayerGB,  
    PixelColorFilter\_BayerGR,  
    PixelColorFilter\_BayerBG,  
    NUM\_PIXELCOLORFILTER }
- enum `_spinAdcBitDepthEnums` {  
    AdcBitDepth\_Bit8,  
    AdcBitDepth\_Bit10,  
    AdcBitDepth\_Bit12,  
    AdcBitDepth\_Bit14,  
    NUM\_ADCBITDEPTH }
- enum `_spinDecimationHorizontalModeEnums` {  
    DecimationHorizontalMode\_Discard,  
    NUM\_DECIMATIONHORIZONTALMODE }
- enum `_spinBinningVerticalModeEnums` {  
    BinningVerticalMode\_Sum,  
    BinningVerticalMode\_Average,  
    NUM\_BINNINGVERTICALMODE }
- enum `_spinPixelSizeEnums` {  
    PixelSize\_Bpp1,  
    PixelSize\_Bpp2,  
    PixelSize\_Bpp4,  
    PixelSize\_Bpp8,  
    PixelSize\_Bpp10,  
    PixelSize\_Bpp12,  
    PixelSize\_Bpp14,  
    PixelSize\_Bpp16,  
    PixelSize\_Bpp20,  
    PixelSize\_Bpp24,  
    PixelSize\_Bpp30,  
    PixelSize\_Bpp32,  
    PixelSize\_Bpp36,  
    PixelSize\_Bpp48,  
    PixelSize\_Bpp64,

```
PixelSize_Bpp96,  
NUM_PIXELSIZE }  
  
• enum _spinDecimationSelectorEnums {  
    DecimationSelector_All,  
    DecimationSelector_Sensor,  
    NUM_DECIMATIONSELECTOR }  
  
• enum _spinImageCompressionModeEnums {  
    ImageCompressionMode_Off,  
    ImageCompressionMode_Lossless,  
    NUM_IMAGECOMPRESSIONMODE }  
  
• enum _spinBinningHorizontalModeEnums {  
    BinningHorizontalMode_Sum,  
    BinningHorizontalMode_Average,  
    NUM_BINNINGHORIZONTALMODE }  
  
• enum _spinPixelFormatEnums {  
    PixelFormat_Mono8,  
    PixelFormat_Mono16,  
    PixelFormat_RGB8Packed,  
    PixelFormat_BayerGR8,  
    PixelFormat_BayerRG8,  
    PixelFormat_BayerGB8,  
    PixelFormat_BayerBG8,  
    PixelFormat_BayerGR16,  
    PixelFormat_BayerRG16,  
    PixelFormat_BayerGB16,  
    PixelFormat_BayerBG16,  
    PixelFormat_Mono12Packed,  
    PixelFormat_BayerGR12Packed,  
    PixelFormat_BayerRG12Packed,  
    PixelFormat_BayerGB12Packed,  
    PixelFormat_BayerBG12Packed,  
    PixelFormat_YUV411Packed,  
    PixelFormat_YUV422Packed,  
    PixelFormat_YUV444Packed,  
    PixelFormat_Mono12p,  
    PixelFormat_BayerGR12p,  
    PixelFormat_BayerRG12p,  
    PixelFormat_BayerGB12p,  
    PixelFormat_BayerBG12p,  
    PixelFormat_YCbCr8,  
    PixelFormat_YCbCr422_8,  
    PixelFormat_YCbCr411_8,  
    PixelFormat_BGR8,  
    PixelFormat_BGRa8,  
    PixelFormat_Mono10Packed,  
    PixelFormat_BayerGR10Packed,  
    PixelFormat_BayerRG10Packed,  
    PixelFormat_BayerGB10Packed,  
    PixelFormat_BayerBG10Packed,  
    PixelFormat_Mono10p,  
    PixelFormat_BayerGR10p,  
    PixelFormat_BayerRG10p,  
    PixelFormat_BayerGB10p,  
    PixelFormat_BayerBG10p,  
    PixelFormat_Mono1p,  
    PixelFormat_Mono2p,  
    PixelFormat_Mono4p,  
    PixelFormat_Mono8s,
```

PixelFormat\_Mono10,  
PixelFormat\_Mono12,  
PixelFormat\_Mono14,  
PixelFormat\_Mono16s,  
PixelFormat\_Mono32f,  
PixelFormat\_BayerBG10,  
PixelFormat\_BayerBG12,  
PixelFormat\_BayerGB10,  
PixelFormat\_BayerGB12,  
PixelFormat\_BayerGR10,  
PixelFormat\_BayerGR12,  
PixelFormat\_BayerRG10,  
PixelFormat\_BayerRG12,  
PixelFormat\_RGBa8,  
PixelFormat\_RGBa10,  
PixelFormat\_RGBa10p,  
PixelFormat\_RGBa12,  
PixelFormat\_RGBa12p,  
PixelFormat\_RGBa14,  
PixelFormat\_RGBa16,  
PixelFormat\_RGB8,  
PixelFormat\_RGB8\_Planar,  
PixelFormat\_RGB10,  
PixelFormat\_RGB10\_Planar,  
PixelFormat\_RGB10p,  
PixelFormat\_RGB10p32,  
PixelFormat\_RGB12,  
PixelFormat\_RGB12\_Planar,  
PixelFormat\_RGB12p,  
PixelFormat\_RGB14,  
PixelFormat\_RGB16,  
PixelFormat\_RGB16s,  
PixelFormat\_RGB32f,  
PixelFormat\_RGB16\_Planar,  
PixelFormat\_RGB565p,  
PixelFormat\_BGRa10,  
PixelFormat\_BGRa10p,  
PixelFormat\_BGRa12,  
PixelFormat\_BGRa12p,  
PixelFormat\_BGRa14,  
PixelFormat\_BGRa16,  
PixelFormat\_RGBa32f,  
PixelFormat\_BGR10,  
PixelFormat\_BGR10p,  
PixelFormat\_BGR12,  
PixelFormat\_BGR12p,  
PixelFormat\_BGR14,  
PixelFormat\_BGR16,  
PixelFormat\_BGR565p,  
PixelFormat\_R8,  
PixelFormat\_R10,  
PixelFormat\_R12,  
PixelFormat\_R16,  
PixelFormat\_G8,  
PixelFormat\_G10,  
PixelFormat\_G12,  
PixelFormat\_G16,  
PixelFormat\_B8,

PixelFormat\_B10,  
PixelFormat\_B12,  
PixelFormat\_B16,  
PixelFormat\_Coord3D\_ABC8,  
PixelFormat\_Coord3D\_ABC8\_Planar,  
PixelFormat\_Coord3D\_ABC10p,  
PixelFormat\_Coord3D\_ABC10p\_Planar,  
PixelFormat\_Coord3D\_ABC12p,  
PixelFormat\_Coord3D\_ABC12p\_Planar,  
PixelFormat\_Coord3D\_ABC16,  
PixelFormat\_Coord3D\_ABC16\_Planar,  
PixelFormat\_Coord3D\_ABC32f,  
PixelFormat\_Coord3D\_ABC32f\_Planar,  
PixelFormat\_Coord3D\_AC8,  
PixelFormat\_Coord3D\_AC8\_Planar,  
PixelFormat\_Coord3D\_AC10p,  
PixelFormat\_Coord3D\_AC10p\_Planar,  
PixelFormat\_Coord3D\_AC12p,  
PixelFormat\_Coord3D\_AC12p\_Planar,  
PixelFormat\_Coord3D\_AC16,  
PixelFormat\_Coord3D\_AC16\_Planar,  
PixelFormat\_Coord3D\_AC32f,  
PixelFormat\_Coord3D\_AC32f\_Planar,  
PixelFormat\_Coord3D\_A8,  
PixelFormat\_Coord3D\_A10p,  
PixelFormat\_Coord3D\_A12p,  
PixelFormat\_Coord3D\_A16,  
PixelFormat\_Coord3D\_A32f,  
PixelFormat\_Coord3D\_B8,  
PixelFormat\_Coord3D\_B10p,  
PixelFormat\_Coord3D\_B12p,  
PixelFormat\_Coord3D\_B16,  
PixelFormat\_Coord3D\_B32f,  
PixelFormat\_Coord3D\_C8,  
PixelFormat\_Coord3D\_C10p,  
PixelFormat\_Coord3D\_C12p,  
PixelFormat\_Coord3D\_C16,  
PixelFormat\_Coord3D\_C32f,  
PixelFormat\_Confidence1,  
PixelFormat\_Confidence1p,  
PixelFormat\_Confidence8,  
PixelFormat\_Confidence16,  
PixelFormat\_Confidence32f,  
PixelFormat\_BiColorBGRG8,  
PixelFormat\_BiColorBGRG10,  
PixelFormat\_BiColorBGRG10p,  
PixelFormat\_BiColorBGRG12,  
PixelFormat\_BiColorBGRG12p,  
PixelFormat\_BiColorRGBG8,  
PixelFormat\_BiColorRGBG10,  
PixelFormat\_BiColorRGBG10p,  
PixelFormat\_BiColorRGBG12,  
PixelFormat\_BiColorRGBG12p,  
PixelFormat\_SCF1WBWG8,  
PixelFormat\_SCF1WBWG10,  
PixelFormat\_SCF1WBWG10p,  
PixelFormat\_SCF1WBWG12,  
PixelFormat\_SCF1WBWG12p,

PixelFormat\_SCF1WBWG14,  
PixelFormat\_SCF1WBWG16,  
PixelFormat\_SCF1WGWB8,  
PixelFormat\_SCF1WGWB10,  
PixelFormat\_SCF1WGWB10p,  
PixelFormat\_SCF1WGWB12,  
PixelFormat\_SCF1WGWB12p,  
PixelFormat\_SCF1WGWB14,  
PixelFormat\_SCF1WGWB16,  
PixelFormat\_SCF1WGWR8,  
PixelFormat\_SCF1WGWR10,  
PixelFormat\_SCF1WGWR10p,  
PixelFormat\_SCF1WGWR12,  
PixelFormat\_SCF1WGWR12p,  
PixelFormat\_SCF1WGWR14,  
PixelFormat\_SCF1WGWR16,  
PixelFormat\_SCF1WRWG8,  
PixelFormat\_SCF1WRWG10,  
PixelFormat\_SCF1WRWG10p,  
PixelFormat\_SCF1WRWG12,  
PixelFormat\_SCF1WRWG12p,  
PixelFormat\_SCF1WRWG14,  
PixelFormat\_SCF1WRWG16,  
PixelFormat\_YCbCr8\_CbYCr,  
PixelFormat\_YCbCr10\_CbYCr,  
PixelFormat\_YCbCr10p\_CbYCr,  
PixelFormat\_YCbCr12\_CbYCr,  
PixelFormat\_YCbCr12p\_CbYCr,  
PixelFormat\_YCbCr411\_8\_CbYYCrYY,  
PixelFormat\_YCbCr422\_8\_CbYCrY,  
PixelFormat\_YCbCr422\_10,  
PixelFormat\_YCbCr422\_10\_CbYCrY,  
PixelFormat\_YCbCr422\_10p,  
PixelFormat\_YCbCr422\_10p\_CbYCrY,  
PixelFormat\_YCbCr422\_12,  
PixelFormat\_YCbCr422\_12\_CbYCrY,  
PixelFormat\_YCbCr422\_12p,  
PixelFormat\_YCbCr422\_12p\_CbYCrY,  
PixelFormat\_YCbCr601\_8\_CbYCr,  
PixelFormat\_YCbCr601\_10\_CbYCr,  
PixelFormat\_YCbCr601\_10p\_CbYCr,  
PixelFormat\_YCbCr601\_12\_CbYCr,  
PixelFormat\_YCbCr601\_12p\_CbYCr,  
PixelFormat\_YCbCr601\_411\_8\_CbYYCrYY,  
PixelFormat\_YCbCr601\_422\_8,  
PixelFormat\_YCbCr601\_422\_8\_CbYCrY,  
PixelFormat\_YCbCr601\_422\_10,  
PixelFormat\_YCbCr601\_422\_10\_CbYCrY,  
PixelFormat\_YCbCr601\_422\_10p,  
PixelFormat\_YCbCr601\_422\_10p\_CbYCrY,  
PixelFormat\_YCbCr601\_422\_12,  
PixelFormat\_YCbCr601\_422\_12\_CbYCrY,  
PixelFormat\_YCbCr601\_422\_12p,  
PixelFormat\_YCbCr601\_422\_12p\_CbYCrY,  
PixelFormat\_YCbCr709\_8\_CbYCr,  
PixelFormat\_YCbCr709\_10\_CbYCr,  
PixelFormat\_YCbCr709\_10p\_CbYCr,  
PixelFormat\_YCbCr709\_12\_CbYCr,

```

PixelFormat_YCbCr709_12p_CbYCr,
PixelFormat_YCbCr709_411_8_CbYYCrYY,
PixelFormat_YCbCr709_422_8,
PixelFormat_YCbCr709_422_8_CbYCrY,
PixelFormat_YCbCr709_422_10,
PixelFormat_YCbCr709_422_10_CbYCrY,
PixelFormat_YCbCr709_422_10p,
PixelFormat_YCbCr709_422_10p_CbYCrY,
PixelFormat_YCbCr709_422_12,
PixelFormat_YCbCr709_422_12_CbYCrY,
PixelFormat_YCbCr709_422_12p,
PixelFormat_YCbCr709_422_12p_CbYCrY,
PixelFormat_YUV8_UYV,
PixelFormat_YUV411_8_UYYVYY,
PixelFormat_YUV422_8,
PixelFormat_YUV422_8_UYVY,
PixelFormat_Polarized8,
PixelFormat_Polarized10p,
PixelFormat_Polarized12p,
PixelFormat_Polarized16,
PixelFormat_BayerRGPolarized8,
PixelFormat_BayerRGPolarized10p,
PixelFormat_BayerRGPolarized12p,
PixelFormat_BayerRGPolarized16,
PixelFormat_LLCMono8,
PixelFormat_LLCBayerRG8,
PixelFormat_JPEGMono8,
PixelFormat_JPEGColor8,
PixelFormat_Raw16,
PixelFormat_Raw8,
PixelFormat_R12_Jpeg,
PixelFormat_GR12_Jpeg,
PixelFormat_GB12_Jpeg,
PixelFormat_B12_Jpeg,
UNKNOWN_PIXELFORMAT,
NUM_PIXELFORMAT }

• enum _spinDecimationVerticalModeEnums {
    DecimationVerticalMode_Discard,
    NUM_DECIMATIONVERTICALMODE }

• enum _spinLineModeEnums {
    LineMode_Input,
    LineMode_Output,
    NUM_LINEMODE }

• enum _spinLineSourceEnums {
    LineSource_Off,
    LineSource_Line0,
    LineSource_Line1,
    LineSource_Line2,
    LineSource_Line3,
    LineSource_UserOutput0,
    LineSource_UserOutput1,
    LineSource_UserOutput2,
    LineSource_UserOutput3,
    LineSource_Counter0Active,
    LineSource_Counter1Active,
    LineSource_LogicBlock0,
    LineSource_LogicBlock1,
    LineSource_ExposureActive,

```



```
LineSource_FrameTriggerWait,  
LineSource_SerialPort0,  
LineSource_PPSSignal,  
LineSource_AllPixel,  
LineSource_AnyPixel,  
NUM_LINESOURCE }  
• enum _spinLineInputFilterSelectorEnums {  
    LineInputFilterSelector_Deglintch,  
    LineInputFilterSelector_Debounce,  
    NUM_LINEINPUTFILTERSELECTOR }  
• enum _spinUserOutputSelectorEnums {  
    UserOutputSelector_UserOutput0,  
    UserOutputSelector_UserOutput1,  
    UserOutputSelector_UserOutput2,  
    UserOutputSelector_UserOutput3,  
    NUM_USEROUTPUTSELECTOR }  
• enum _spinLineFormatEnums {  
    LineFormat_NoConnect,  
    LineFormat_TriState,  
    LineFormat_TTL,  
    LineFormat_LVDS,  
    LineFormat_RS422,  
    LineFormat_OptoCoupled,  
    LineFormat_OpenDrain,  
    NUM_LINEFORMAT }  
• enum _spinLineSelectorEnums {  
    LineSelector_Line0,  
    LineSelector_Line1,  
    LineSelector_Line2,  
    LineSelector_Line3,  
    NUM_LINESELECTOR }  
• enum _spinExposureActiveModeEnums {  
    ExposureActiveMode_Line1,  
    ExposureActiveMode_AnyPixels,  
    ExposureActiveMode_AllPixels,  
    NUM_EXPOSUREACTIVEMODE }  
• enum _spinCounterTriggerActivationEnums {  
    CounterTriggerActivation_LevelLow,  
    CounterTriggerActivation_LevelHigh,  
    CounterTriggerActivation_FallingEdge,  
    CounterTriggerActivation_RisingEdge,  
    CounterTriggerActivation_AnyEdge,  
    NUM_COUNTERTRIGGERACTIVATION }  
• enum _spinCounterSelectorEnums {  
    CounterSelector_Counter0,  
    CounterSelector_Counter1,  
    NUM_COUNTERSELECTOR }  
• enum _spinCounterStatusEnums {  
    CounterStatus_CounterIdle,  
    CounterStatus_CounterTriggerWait,  
    CounterStatus_CounterActive,  
    CounterStatus_CounterCompleted,  
    CounterStatus_CounterOverflow,  
    NUM_COUNTERSTATUS }  
• enum _spinCounterTriggerSourceEnums {  
    CounterTriggerSource_Off,  
    CounterTriggerSource_Line0,  
    CounterTriggerSource_Line1,
```

```

CounterTriggerSource_Line2,
CounterTriggerSource_Line3,
CounterTriggerSource_UserOutput0,
CounterTriggerSource_UserOutput1,
CounterTriggerSource_UserOutput2,
CounterTriggerSource_UserOutput3,
CounterTriggerSource_Counter0Start,
CounterTriggerSource_Counter1Start,
CounterTriggerSource_Counter0End,
CounterTriggerSource_Counter1End,
CounterTriggerSource_LogicBlock0,
CounterTriggerSource_LogicBlock1,
CounterTriggerSource_ExposureStart,
CounterTriggerSource_ExposureEnd,
CounterTriggerSource_FrameTriggerWait,
NUM_COUNTERTRIGGERSOURCE }

• enum _spinCounterResetSourceEnums {
CounterResetSource_Off,
CounterResetSource_Line0,
CounterResetSource_Line1,
CounterResetSource_Line2,
CounterResetSource_Line3,
CounterResetSource_UserOutput0,
CounterResetSource_UserOutput1,
CounterResetSource_UserOutput2,
CounterResetSource_UserOutput3,
CounterResetSource_Counter0Start,
CounterResetSource_Counter1Start,
CounterResetSource_Counter0End,
CounterResetSource_Counter1End,
CounterResetSource_LogicBlock0,
CounterResetSource_LogicBlock1,
CounterResetSource_ExposureStart,
CounterResetSource_ExposureEnd,
CounterResetSource_FrameTriggerWait,
NUM_COUNTERRESETSOURCE }

• enum _spinCounterEventSourceEnums {
CounterEventSource_Off,
CounterEventSource_MHzTick,
CounterEventSource_Line0,
CounterEventSource_Line1,
CounterEventSource_Line2,
CounterEventSource_Line3,
CounterEventSource_UserOutput0,
CounterEventSource_UserOutput1,
CounterEventSource_UserOutput2,
CounterEventSource_UserOutput3,
CounterEventSource_Counter0Start,
CounterEventSource_Counter1Start,
CounterEventSource_Counter0End,
CounterEventSource_Counter1End,
CounterEventSource_LogicBlock0,
CounterEventSource_LogicBlock1,
CounterEventSource_ExposureStart,
CounterEventSource_ExposureEnd,
CounterEventSource_FrameTriggerWait,
NUM_COUNTEREVENTSOURCE }

• enum _spinCounterEventActivationEnums {

```

```

CounterEventActivation_LevelLow,
CounterEventActivation_LevelHigh,
CounterEventActivation_FallingEdge,
CounterEventActivation_RisingEdge,
CounterEventActivation_AnyEdge,
NUM_COUNTEREVENTACTIVATION }

• enum _spinCounterResetActivationEnums {
CounterResetActivation_LevelLow,
CounterResetActivation_LevelHigh,
CounterResetActivation_FallingEdge,
CounterResetActivation_RisingEdge,
CounterResetActivation_AnyEdge,
NUM_COUNTERRESETACTIVATION }

• enum _spinDeviceTypeEnums {
DeviceType_Transmitter,
DeviceType_Receiver,
DeviceType_Transceiver,
DeviceType_Peripheral,
NUM_DEVICETYPE }

• enum _spinDeviceConnectionStatusEnums {
DeviceConnectionStatus_Active,
DeviceConnectionStatus_Inactive,
NUM_DEVICECONNECTIONSTATUS }

• enum _spinDeviceLinkThroughputLimitModeEnums {
DeviceLinkThroughputLimitMode_On,
DeviceLinkThroughputLimitMode_Off,
NUM_DEVICELINKTHROUGHPUTLIMITMODE }

• enum _spinDeviceLinkHeartbeatModeEnums {
DeviceLinkHeartbeatMode_On,
DeviceLinkHeartbeatMode_Off,
NUM_DEVICELINKHEARTBEATMODE }

• enum _spinDeviceStreamChannelTypeEnums {
DeviceStreamChannelType_Transmitter,
DeviceStreamChannelType_Receiver,
NUM_DEVICESTREAMCHANNELTYPE }

• enum _spinDeviceStreamChannelEndiannessEnums {
DeviceStreamChannelEndianness_Big,
DeviceStreamChannelEndianness_Little,
NUM_DEVICESTREAMCHANNELENDIANNESS }

• enum _spinDeviceClockSelectorEnums {
DeviceClockSelector_Sensor,
DeviceClockSelector_SensorDigitization,
DeviceClockSelector_CameraLink,
NUM_DEVICECLOCKSELECTOR }

• enum _spinDeviceSerialPortSelectorEnums {
DeviceSerialPortSelector_CameraLink,
NUM_DEVICESERIALPORTSELECTOR }

• enum _spinDeviceSerialPortBaudRateEnums {
DeviceSerialPortBaudRate_Baud9600,
DeviceSerialPortBaudRate_Baud19200,
DeviceSerialPortBaudRate_Baud38400,
DeviceSerialPortBaudRate_Baud57600,
DeviceSerialPortBaudRate_Baud115200,
DeviceSerialPortBaudRate_Baud230400,
DeviceSerialPortBaudRate_Baud460800,
DeviceSerialPortBaudRate_Baud921600,
NUM_DEVICESERIALPORTBAUDRATE }

```

- `enum _spinSensorTapsEnums {  
SensorTaps_One,  
SensorTaps_Two,  
SensorTaps_Three,  
SensorTaps_Four,  
SensorTaps_Eight,  
SensorTaps_Ten,  
NUM_SENSORTAPS }`
- `enum _spinSensorDigitizationTapsEnums {  
SensorDigitizationTaps_One,  
SensorDigitizationTaps_Two,  
SensorDigitizationTaps_Three,  
SensorDigitizationTaps_Four,  
SensorDigitizationTaps_Eight,  
SensorDigitizationTaps_Ten,  
NUM_SENSORDIGITIZATIONTAPS }`
- `enum _spinRegionSelectorEnums {  
RegionSelector_Region0,  
RegionSelector_Region1,  
RegionSelector_Region2,  
RegionSelector_All,  
NUM_REGIONSELECTOR }`
- `enum _spinRegionModeEnums {  
RegionMode_Off,  
RegionMode_On,  
NUM_REGIONMODE }`
- `enum _spinRegionDestinationEnums {  
RegionDestination_Stream0,  
RegionDestination_Stream1,  
RegionDestination_Stream2,  
NUM_REGIONDESTINATION }`
- `enum _spinImageComponentSelectorEnums {  
ImageComponentSelector_Intensity,  
ImageComponentSelector_Color,  
ImageComponentSelector_Infrared,  
ImageComponentSelector_Ultraviolet,  
ImageComponentSelector_Range,  
ImageComponentSelector_Disparity,  
ImageComponentSelector_Confidence,  
ImageComponentSelector_Scatter,  
NUM_IMAGECOMPONENTSELECTOR }`
- `enum _spinPixelFormatInfoSelectorEnums {  
PixelFormatInfoSelector_Mono1p,  
PixelFormatInfoSelector_Mono2p,  
PixelFormatInfoSelector_Mono4p,  
PixelFormatInfoSelector_Mono8,  
PixelFormatInfoSelector_Mono8s,  
PixelFormatInfoSelector_Mono10,  
PixelFormatInfoSelector_Mono10p,  
PixelFormatInfoSelector_Mono12,  
PixelFormatInfoSelector_Mono12p,  
PixelFormatInfoSelector_Mono14,  
PixelFormatInfoSelector_Mono16,  
PixelFormatInfoSelector_Mono16s,  
PixelFormatInfoSelector_Mono32f,  
PixelFormatInfoSelector_BayerBG8,  
PixelFormatInfoSelector_BayerBG10,  
PixelFormatInfoSelector_BayerBG10p,`

[PixelFormatInfoSelector\\_BayerBG12,](#)  
[PixelFormatInfoSelector\\_BayerBG12p,](#)  
[PixelFormatInfoSelector\\_BayerBG16,](#)  
[PixelFormatInfoSelector\\_BayerGB8,](#)  
[PixelFormatInfoSelector\\_BayerGB10,](#)  
[PixelFormatInfoSelector\\_BayerGB10p,](#)  
[PixelFormatInfoSelector\\_BayerGB12,](#)  
[PixelFormatInfoSelector\\_BayerGB12p,](#)  
[PixelFormatInfoSelector\\_BayerGB16,](#)  
[PixelFormatInfoSelector\\_BayerGR8,](#)  
[PixelFormatInfoSelector\\_BayerGR10,](#)  
[PixelFormatInfoSelector\\_BayerGR10p,](#)  
[PixelFormatInfoSelector\\_BayerGR12,](#)  
[PixelFormatInfoSelector\\_BayerGR12p,](#)  
[PixelFormatInfoSelector\\_BayerGR16,](#)  
[PixelFormatInfoSelector\\_BayerRG8,](#)  
[PixelFormatInfoSelector\\_BayerRG10,](#)  
[PixelFormatInfoSelector\\_BayerRG10p,](#)  
[PixelFormatInfoSelector\\_BayerRG12,](#)  
[PixelFormatInfoSelector\\_BayerRG12p,](#)  
[PixelFormatInfoSelector\\_BayerRG16,](#)  
[PixelFormatInfoSelector\\_RGBa8,](#)  
[PixelFormatInfoSelector\\_RGBa10,](#)  
[PixelFormatInfoSelector\\_RGBa10p,](#)  
[PixelFormatInfoSelector\\_RGBa12,](#)  
[PixelFormatInfoSelector\\_RGBa12p,](#)  
[PixelFormatInfoSelector\\_RGBa14,](#)  
[PixelFormatInfoSelector\\_RGBa16,](#)  
[PixelFormatInfoSelector\\_RGB8,](#)  
[PixelFormatInfoSelector\\_RGB8\\_Planar,](#)  
[PixelFormatInfoSelector\\_RGB10,](#)  
[PixelFormatInfoSelector\\_RGB10\\_Planar,](#)  
[PixelFormatInfoSelector\\_RGB10p,](#)  
[PixelFormatInfoSelector\\_RGB10p32,](#)  
[PixelFormatInfoSelector\\_RGB12,](#)  
[PixelFormatInfoSelector\\_RGB12\\_Planar,](#)  
[PixelFormatInfoSelector\\_RGB12p,](#)  
[PixelFormatInfoSelector\\_RGB14,](#)  
[PixelFormatInfoSelector\\_RGB16,](#)  
[PixelFormatInfoSelector\\_RGB16s,](#)  
[PixelFormatInfoSelector\\_RGB32f,](#)  
[PixelFormatInfoSelector\\_RGB16\\_Planar,](#)  
[PixelFormatInfoSelector\\_RGB565p,](#)  
[PixelFormatInfoSelector\\_BGRa8,](#)  
[PixelFormatInfoSelector\\_BGRa10,](#)  
[PixelFormatInfoSelector\\_BGRa10p,](#)  
[PixelFormatInfoSelector\\_BGRa12,](#)  
[PixelFormatInfoSelector\\_BGRa12p,](#)  
[PixelFormatInfoSelector\\_BGRa14,](#)  
[PixelFormatInfoSelector\\_BGRa16,](#)  
[PixelFormatInfoSelector\\_RGBa32f,](#)  
[PixelFormatInfoSelector\\_BGR8,](#)  
[PixelFormatInfoSelector\\_BGR10,](#)  
[PixelFormatInfoSelector\\_BGR10p,](#)  
[PixelFormatInfoSelector\\_BGR12,](#)  
[PixelFormatInfoSelector\\_BGR12p,](#)  
[PixelFormatInfoSelector\\_BGR14,](#)  
[PixelFormatInfoSelector\\_BGR16,](#)

PixelFormatInfoSelector\_BGR565p,  
PixelFormatInfoSelector\_R8,  
PixelFormatInfoSelector\_R10,  
PixelFormatInfoSelector\_R12,  
PixelFormatInfoSelector\_R16,  
PixelFormatInfoSelector\_G8,  
PixelFormatInfoSelector\_G10,  
PixelFormatInfoSelector\_G12,  
PixelFormatInfoSelector\_G16,  
PixelFormatInfoSelector\_B8,  
PixelFormatInfoSelector\_B10,  
PixelFormatInfoSelector\_B12,  
PixelFormatInfoSelector\_B16,  
PixelFormatInfoSelector\_Coord3D\_ABC8,  
PixelFormatInfoSelector\_Coord3D\_ABC8\_Planar,  
PixelFormatInfoSelector\_Coord3D\_ABC10p,  
PixelFormatInfoSelector\_Coord3D\_ABC10p\_Planar,  
PixelFormatInfoSelector\_Coord3D\_ABC12p,  
PixelFormatInfoSelector\_Coord3D\_ABC12p\_Planar,  
PixelFormatInfoSelector\_Coord3D\_ABC16,  
PixelFormatInfoSelector\_Coord3D\_ABC16\_Planar,  
PixelFormatInfoSelector\_Coord3D\_ABC32f,  
PixelFormatInfoSelector\_Coord3D\_ABC32f\_Planar,  
PixelFormatInfoSelector\_Coord3D\_AC8,  
PixelFormatInfoSelector\_Coord3D\_AC8\_Planar,  
PixelFormatInfoSelector\_Coord3D\_AC10p,  
PixelFormatInfoSelector\_Coord3D\_AC10p\_Planar,  
PixelFormatInfoSelector\_Coord3D\_AC12p,  
PixelFormatInfoSelector\_Coord3D\_AC12p\_Planar,  
PixelFormatInfoSelector\_Coord3D\_AC16,  
PixelFormatInfoSelector\_Coord3D\_AC16\_Planar,  
PixelFormatInfoSelector\_Coord3D\_AC32f,  
PixelFormatInfoSelector\_Coord3D\_AC32f\_Planar,  
PixelFormatInfoSelector\_Coord3D\_A8,  
PixelFormatInfoSelector\_Coord3D\_A10p,  
PixelFormatInfoSelector\_Coord3D\_A12p,  
PixelFormatInfoSelector\_Coord3D\_A16,  
PixelFormatInfoSelector\_Coord3D\_A32f,  
PixelFormatInfoSelector\_Coord3D\_B8,  
PixelFormatInfoSelector\_Coord3D\_B10p,  
PixelFormatInfoSelector\_Coord3D\_B12p,  
PixelFormatInfoSelector\_Coord3D\_B16,  
PixelFormatInfoSelector\_Coord3D\_B32f,  
PixelFormatInfoSelector\_Coord3D\_C8,  
PixelFormatInfoSelector\_Coord3D\_C10p,  
PixelFormatInfoSelector\_Coord3D\_C12p,  
PixelFormatInfoSelector\_Coord3D\_C16,  
PixelFormatInfoSelector\_Coord3D\_C32f,  
PixelFormatInfoSelector\_Confidence1,  
PixelFormatInfoSelector\_Confidence1p,  
PixelFormatInfoSelector\_Confidence8,  
PixelFormatInfoSelector\_Confidence16,  
PixelFormatInfoSelector\_Confidence32f,  
PixelFormatInfoSelector\_BiColorBGRG8,  
PixelFormatInfoSelector\_BiColorBGRG10,  
PixelFormatInfoSelector\_BiColorBGRG10p,  
PixelFormatInfoSelector\_BiColorBGRG12,  
PixelFormatInfoSelector\_BiColorBGRG12p,

[PixelFormatInfoSelector\\_BiColorRGBG8,](#)  
[PixelFormatInfoSelector\\_BiColorRGBG10,](#)  
[PixelFormatInfoSelector\\_BiColorRGBG10p,](#)  
[PixelFormatInfoSelector\\_BiColorRGBG12,](#)  
[PixelFormatInfoSelector\\_BiColorRGBG12p,](#)  
[PixelFormatInfoSelector\\_SCF1WBWG8,](#)  
[PixelFormatInfoSelector\\_SCF1WBWG10,](#)  
[PixelFormatInfoSelector\\_SCF1WBWG10p,](#)  
[PixelFormatInfoSelector\\_SCF1WBWG12,](#)  
[PixelFormatInfoSelector\\_SCF1WBWG12p,](#)  
[PixelFormatInfoSelector\\_SCF1WBWG14,](#)  
[PixelFormatInfoSelector\\_SCF1WBWG16,](#)  
[PixelFormatInfoSelector\\_SCF1WGWB8,](#)  
[PixelFormatInfoSelector\\_SCF1WGWB10,](#)  
[PixelFormatInfoSelector\\_SCF1WGWB10p,](#)  
[PixelFormatInfoSelector\\_SCF1WGWB12,](#)  
[PixelFormatInfoSelector\\_SCF1WGWB12p,](#)  
[PixelFormatInfoSelector\\_SCF1WGWB14,](#)  
[PixelFormatInfoSelector\\_SCF1WGWB16,](#)  
[PixelFormatInfoSelector\\_SCF1WGWR8,](#)  
[PixelFormatInfoSelector\\_SCF1WGWR10,](#)  
[PixelFormatInfoSelector\\_SCF1WGWR10p,](#)  
[PixelFormatInfoSelector\\_SCF1WGWR12,](#)  
[PixelFormatInfoSelector\\_SCF1WGWR12p,](#)  
[PixelFormatInfoSelector\\_SCF1WGWR14,](#)  
[PixelFormatInfoSelector\\_SCF1WGWR16,](#)  
[PixelFormatInfoSelector\\_SCF1WRWG8,](#)  
[PixelFormatInfoSelector\\_SCF1WRWG10,](#)  
[PixelFormatInfoSelector\\_SCF1WRWG10p,](#)  
[PixelFormatInfoSelector\\_SCF1WRWG12,](#)  
[PixelFormatInfoSelector\\_SCF1WRWG12p,](#)  
[PixelFormatInfoSelector\\_SCF1WRWG14,](#)  
[PixelFormatInfoSelector\\_SCF1WRWG16,](#)  
[PixelFormatInfoSelector\\_YCbCr8,](#)  
[PixelFormatInfoSelector\\_YCbCr8\\_CbYCr,](#)  
[PixelFormatInfoSelector\\_YCbCr10\\_CbYCr,](#)  
[PixelFormatInfoSelector\\_YCbCr10p\\_CbYCr,](#)  
[PixelFormatInfoSelector\\_YCbCr12\\_CbYCr,](#)  
[PixelFormatInfoSelector\\_YCbCr12p\\_CbYCr,](#)  
[PixelFormatInfoSelector\\_YCbCr411\\_8,](#)  
[PixelFormatInfoSelector\\_YCbCr411\\_8\\_CbYYCrYY,](#)  
[PixelFormatInfoSelector\\_YCbCr422\\_8,](#)  
[PixelFormatInfoSelector\\_YCbCr422\\_8\\_CbYCrY,](#)  
[PixelFormatInfoSelector\\_YCbCr422\\_10,](#)  
[PixelFormatInfoSelector\\_YCbCr422\\_10\\_CbYCrY,](#)  
[PixelFormatInfoSelector\\_YCbCr422\\_10p,](#)  
[PixelFormatInfoSelector\\_YCbCr422\\_10p\\_CbYCrY,](#)  
[PixelFormatInfoSelector\\_YCbCr422\\_12,](#)  
[PixelFormatInfoSelector\\_YCbCr422\\_12\\_CbYCrY,](#)  
[PixelFormatInfoSelector\\_YCbCr422\\_12p,](#)  
[PixelFormatInfoSelector\\_YCbCr422\\_12p\\_CbYCrY,](#)  
[PixelFormatInfoSelector\\_YCbCr601\\_8\\_CbYCr,](#)  
[PixelFormatInfoSelector\\_YCbCr601\\_10\\_CbYCr,](#)  
[PixelFormatInfoSelector\\_YCbCr601\\_10p\\_CbYCr,](#)  
[PixelFormatInfoSelector\\_YCbCr601\\_12\\_CbYCr,](#)  
[PixelFormatInfoSelector\\_YCbCr601\\_12p\\_CbYCr,](#)  
[PixelFormatInfoSelector\\_YCbCr601\\_411\\_8\\_CbYYCrYY,](#)  
[PixelFormatInfoSelector\\_YCbCr601\\_422\\_8,](#)

```

PixelFormatInfoSelector_YCbCr601_422_8_CbYCrY,
PixelFormatInfoSelector_YCbCr601_422_10,
PixelFormatInfoSelector_YCbCr601_422_10_CbYCrY,
PixelFormatInfoSelector_YCbCr601_422_10p,
PixelFormatInfoSelector_YCbCr601_422_10p_CbYCrY,
PixelFormatInfoSelector_YCbCr601_422_12,
PixelFormatInfoSelector_YCbCr601_422_12_CbYCrY,
PixelFormatInfoSelector_YCbCr601_422_12p,
PixelFormatInfoSelector_YCbCr601_422_12p_CbYCrY,
PixelFormatInfoSelector_YCbCr709_8_CbYCr,
PixelFormatInfoSelector_YCbCr709_10_CbYCr,
PixelFormatInfoSelector_YCbCr709_10p_CbYCr,
PixelFormatInfoSelector_YCbCr709_12_CbYCr,
PixelFormatInfoSelector_YCbCr709_12p_CbYCr,
PixelFormatInfoSelector_YCbCr709_411_8_CbYYCrYY,
PixelFormatInfoSelector_YCbCr709_422_8,
PixelFormatInfoSelector_YCbCr709_422_8_CbYCrY,
PixelFormatInfoSelector_YCbCr709_422_10,
PixelFormatInfoSelector_YCbCr709_422_10_CbYCrY,
PixelFormatInfoSelector_YCbCr709_422_10p,
PixelFormatInfoSelector_YCbCr709_422_10p_CbYCrY,
PixelFormatInfoSelector_YCbCr709_422_12,
PixelFormatInfoSelector_YCbCr709_422_12_CbYCrY,
PixelFormatInfoSelector_YCbCr709_422_12p,
PixelFormatInfoSelector_YCbCr709_422_12p_CbYCrY,
PixelFormatInfoSelector_YUV8_UYV,
PixelFormatInfoSelector_YUV411_8_UYYVYY,
PixelFormatInfoSelector_YUV422_8,
PixelFormatInfoSelector_YUV422_8_UYVY,
PixelFormatInfoSelector_Polarized8,
PixelFormatInfoSelector_Polarized10p,
PixelFormatInfoSelector_Polarized12p,
PixelFormatInfoSelector_Polarized16,
PixelFormatInfoSelector_BayerRGPolarized8,
PixelFormatInfoSelector_BayerRGPolarized10p,
PixelFormatInfoSelector_BayerRGPolarized12p,
PixelFormatInfoSelector_BayerRGPolarized16,
PixelFormatInfoSelector_LLCMono8,
PixelFormatInfoSelector_LLCBayerRG8,
PixelFormatInfoSelector_JPEGMono8,
PixelFormatInfoSelector_JPEGColor8,
NUM_PIXELFORMATINFOSELECTOR }

• enum _spinDeinterlacingEnums {
    Deinterlacing_Off,
    Deinterlacing_LineDuplication,
    Deinterlacing_Weave,
    NUM_DEINTERLACING }

• enum _spinImageCompressionRateOptionEnums {
    ImageCompressionRateOption_FixBitrate,
    ImageCompressionRateOption_FixQuality,
    NUM_IMAGECOMPRESSIONRATEOPTION }

• enum _spinImageCompressionJPEGFormatOptionEnums {
    ImageCompressionJPEGFormatOption_Lossless,
    ImageCompressionJPEGFormatOption_BaselineStandard,
    ImageCompressionJPEGFormatOption_BaselineOptimized,
    ImageCompressionJPEGFormatOption_Progressive,
    NUM_IMAGECOMPRESSIONJPEGFORMATOPTION }

• enum _spinAcquisitionStatusSelectorEnums {

```



```

    AcquisitionStatusSelector_AcquisitionTriggerWait,
    AcquisitionStatusSelector_AcquisitionActive,
    AcquisitionStatusSelector_AcquisitionTransfer,
    AcquisitionStatusSelector_FrameTriggerWait,
    AcquisitionStatusSelector_FrameActive,
    AcquisitionStatusSelector_ExposureActive,
    NUM_ACQUISITIONSTATUSSELECTOR }

• enum _spinExposureTimeModeEnums {
    ExposureTimeMode_Common,
    ExposureTimeMode_Individual,
    NUM_EXPOSURETIMEMODE }

• enum _spinExposureTimeSelectorEnums {
    ExposureTimeSelector_Common,
    ExposureTimeSelector_Red,
    ExposureTimeSelector_Green,
    ExposureTimeSelector_Blue,
    ExposureTimeSelector_Cyan,
    ExposureTimeSelector_Magenta,
    ExposureTimeSelector_Yellow,
    ExposureTimeSelector_Infrared,
    ExposureTimeSelector_Ultraviolet,
    ExposureTimeSelector_Stage1,
    ExposureTimeSelector_Stage2,
    NUM_EXPOSURETIMESELECTOR }

• enum _spinGainAutoBalanceEnums {
    GainAutoBalance_Off,
    GainAutoBalance_Once,
    GainAutoBalance_Continuous,
    NUM_GAINAUTOBALANCE }

• enum _spinBlackLevelAutoEnums {
    BlackLevelAuto_Off,
    BlackLevelAuto_Once,
    BlackLevelAuto_Continuous,
    NUM_BLACKLEVELAUTO }

• enum _spinBlackLevelAutoBalanceEnums {
    BlackLevelAutoBalance_Off,
    BlackLevelAutoBalance_Once,
    BlackLevelAutoBalance_Continuous,
    NUM_BLACKLEVELAUTOBALANCE }

• enum _spinWhiteClipSelectorEnums {
    WhiteClipSelector_All,
    WhiteClipSelector_Red,
    WhiteClipSelector_Green,
    WhiteClipSelector_Blue,
    WhiteClipSelector_Y,
    WhiteClipSelector_U,
    WhiteClipSelector_V,
    WhiteClipSelector_Tap1,
    WhiteClipSelector_Tap2,
    NUM_WHITECLIPSELECTOR }

• enum _spinTimerSelectorEnums {
    TimerSelector_Timer0,
    TimerSelector_Timer1,
    TimerSelector_Timer2,
    NUM_TIMERSELECTOR }

• enum _spinTimerStatusEnums {
    TimerStatus_TimerIdle,
    TimerStatus_TimerTriggerWait,

```

```

TimerStatus_TimerActive,
TimerStatus_TimerCompleted,
NUM_TIMERSTATUS }

• enum _spinTimerTriggerSourceEnums {
TimerTriggerSource_Off,
TimerTriggerSource_AcquisitionTrigger,
TimerTriggerSource_AcquisitionStart,
TimerTriggerSource_AcquisitionEnd,
TimerTriggerSource_FrameTrigger,
TimerTriggerSource_FrameStart,
TimerTriggerSource_FrameEnd,
TimerTriggerSource_FrameBurstStart,
TimerTriggerSource_FrameBurstEnd,
TimerTriggerSource_LineTrigger,
TimerTriggerSource_LineStart,
TimerTriggerSource_LineEnd,
TimerTriggerSource_ExposureStart,
TimerTriggerSource_ExposureEnd,
TimerTriggerSource_Line0,
TimerTriggerSource_Line1,
TimerTriggerSource_Line2,
TimerTriggerSource_UserOutput0,
TimerTriggerSource_UserOutput1,
TimerTriggerSource_UserOutput2,
TimerTriggerSource_Counter0Start,
TimerTriggerSource_Counter1Start,
TimerTriggerSource_Counter2Start,
TimerTriggerSource_Counter0End,
TimerTriggerSource_Counter1End,
TimerTriggerSource_Counter2End,
TimerTriggerSource_Timer0Start,
TimerTriggerSource_Timer1Start,
TimerTriggerSource_Timer2Start,
TimerTriggerSource_Timer0End,
TimerTriggerSource_Timer1End,
TimerTriggerSource_Timer2End,
TimerTriggerSource_Encoder0,
TimerTriggerSource_Encoder1,
TimerTriggerSource_Encoder2,
TimerTriggerSource_SoftwareSignal0,
TimerTriggerSource_SoftwareSignal1,
TimerTriggerSource_SoftwareSignal2,
TimerTriggerSource_Action0,
TimerTriggerSource_Action1,
TimerTriggerSource_Action2,
TimerTriggerSource_LinkTrigger0,
TimerTriggerSource_LinkTrigger1,
TimerTriggerSource_LinkTrigger2,
NUM_TIMERTRIGGERSOURCE }

• enum _spinTimerTriggerActivationEnums {
TimerTriggerActivation_RisingEdge,
TimerTriggerActivation_FallingEdge,
TimerTriggerActivation_AnyEdge,
TimerTriggerActivation_LevelHigh,
TimerTriggerActivation_LevelLow,
NUM_TIMERTRIGGERACTIVATION }

• enum _spinEncoderSelectorEnums {
EncoderSelector_Encoder0,

```

```

EncoderSelector_Encoder1,
EncoderSelector_Encoder2,
NUM_ENCODERSELECTOR }

• enum _spinEncoderSourceAEnums {
EncoderSourceA_Off,
EncoderSourceA_Line0,
EncoderSourceA_Line1,
EncoderSourceA_Line2,
NUM_ENCODERSOURCEA }

• enum _spinEncoderSourceBEnums {
EncoderSourceB_Off,
EncoderSourceB_Line0,
EncoderSourceB_Line1,
EncoderSourceB_Line2,
NUM_ENCODERSOURCEB }

• enum _spinEncoderModeEnums {
EncoderMode_FourPhase,
EncoderMode_HighResolution,
NUM_ENCODERMODE }

• enum _spinEncoderOutputModeEnums {
EncoderOutputMode_Off,
EncoderOutputMode_PositionUp,
EncoderOutputMode_PositionDown,
EncoderOutputMode_DirectionUp,
EncoderOutputMode_DirectionDown,
EncoderOutputMode_Motion,
NUM_ENCODEROUTPUTMODE }

• enum _spinEncoderStatusEnums {
EncoderStatus_EncoderUp,
EncoderStatus_EncoderDown,
EncoderStatus_EncoderIdle,
EncoderStatus_EncoderStatic,
NUM_ENCODERSTATUS }

• enum _spinEncoderResetSourceEnums {
EncoderResetSource_Off,
EncoderResetSource_AcquisitionTrigger,
EncoderResetSource_AcquisitionStart,
EncoderResetSource_AcquisitionEnd,
EncoderResetSource_FrameTrigger,
EncoderResetSource_FrameStart,
EncoderResetSource_FrameEnd,
EncoderResetSource_ExposureStart,
EncoderResetSource_ExposureEnd,
EncoderResetSource_Line0,
EncoderResetSource_Line1,
EncoderResetSource_Line2,
EncoderResetSource_Counter0Start,
EncoderResetSource_Counter1Start,
EncoderResetSource_Counter2Start,
EncoderResetSource_Counter0End,
EncoderResetSource_Counter1End,
EncoderResetSource_Counter2End,
EncoderResetSource_Timer0Start,
EncoderResetSource_Timer1Start,
EncoderResetSource_Timer2Start,
EncoderResetSource_Timer0End,
EncoderResetSource_Timer1End,
EncoderResetSource_Timer2End,

```

```

EncoderResetSource_UserOutput0,
EncoderResetSource_UserOutput1,
EncoderResetSource_UserOutput2,
EncoderResetSource_SoftwareSignal0,
EncoderResetSource_SoftwareSignal1,
EncoderResetSource_SoftwareSignal2,
EncoderResetSource_Action0,
EncoderResetSource_Action1,
EncoderResetSource_Action2,
EncoderResetSource_LinkTrigger0,
EncoderResetSource_LinkTrigger1,
EncoderResetSource_LinkTrigger2,
NUM_ENCODERRESETSOURCE }

• enum _spinEncoderResetActivationEnums {
EncoderResetActivation_RisingEdge,
EncoderResetActivation_FallingEdge,
EncoderResetActivation_AnyEdge,
EncoderResetActivation_LevelHigh,
EncoderResetActivation_LevelLow,
NUM_ENCODERRESETACTIVATION }

• enum _spinSoftwareSignalSelectorEnums {
SoftwareSignalSelector_SoftwareSignal0,
SoftwareSignalSelector_SoftwareSignal1,
SoftwareSignalSelector_SoftwareSignal2,
NUM_SOFTWARESIGNALSELECTOR }

• enum _spinActionUnconditionalModeEnums {
ActionUnconditionalMode_Off,
ActionUnconditionalMode_On,
NUM_ACTIONUNCONDITIONALMODE }

• enum _spinSourceSelectorEnums {
SourceSelector_Source0,
SourceSelector_Source1,
SourceSelector_Source2,
SourceSelector_All,
NUM_SOURCESELECTOR }

• enum _spinTransferSelectorEnums {
TransferSelector_Stream0,
TransferSelector_Stream1,
TransferSelector_Stream2,
TransferSelector_All,
NUM_TRANSFERSELECTOR }

• enum _spinTransferTriggerSelectorEnums {
TransferTriggerSelector_TransferStart,
TransferTriggerSelector_TransferStop,
TransferTriggerSelector_TransferAbort,
TransferTriggerSelector_TransferPause,
TransferTriggerSelector_TransferResume,
TransferTriggerSelector_TransferActive,
TransferTriggerSelector_TransferBurstStart,
TransferTriggerSelector_TransferBurstStop,
NUM_TRANSFERTRIGGERSELECTOR }

• enum _spinTransferTriggerModeEnums {
TransferTriggerMode_Off,
TransferTriggerMode_On,
NUM_TRANSFERTRIGGERMODE }

• enum _spinTransferTriggerSourceEnums {
TransferTriggerSource_Line0,
TransferTriggerSource_Line1,

```

```

TransferTriggerSource_Line2,
TransferTriggerSource_Counter0Start,
TransferTriggerSource_Counter1Start,
TransferTriggerSource_Counter2Start,
TransferTriggerSource_Counter0End,
TransferTriggerSource_Counter1End,
TransferTriggerSource_Counter2End,
TransferTriggerSource_Timer0Start,
TransferTriggerSource_Timer1Start,
TransferTriggerSource_Timer2Start,
TransferTriggerSource_Timer0End,
TransferTriggerSource_Timer1End,
TransferTriggerSource_Timer2End,
TransferTriggerSource_SoftwareSignal0,
TransferTriggerSource_SoftwareSignal1,
TransferTriggerSource_SoftwareSignal2,
TransferTriggerSource_Action0,
TransferTriggerSource_Action1,
TransferTriggerSource_Action2,
NUM_TRANSFERTRIGGERSOURCE }
• enum _spinTransferTriggerActivationEnums {
TransferTriggerActivation_RisingEdge,
TransferTriggerActivation_FallingEdge,
TransferTriggerActivation_AnyEdge,
TransferTriggerActivation_LevelHigh,
TransferTriggerActivation_LevelLow,
NUM_TRANSFERTRIGGERACTIVATION }
• enum _spinTransferStatusSelectorEnums {
TransferStatusSelector_Streaming,
TransferStatusSelector_Paused,
TransferStatusSelector_Stopping,
TransferStatusSelector_Stopped,
TransferStatusSelector_QueueOverflow,
NUM_TRANSFERSTATUSSELECTOR }
• enum _spinTransferComponentSelectorEnums {
TransferComponentSelector_Red,
TransferComponentSelector_Green,
TransferComponentSelector_Blue,
TransferComponentSelector_All,
NUM_TRANSFERCOMPONENTSELECTOR }
• enum _spinScan3dDistanceUnitEnums {
Scan3dDistanceUnit_Millimeter,
Scan3dDistanceUnit_Inch,
NUM_SCAN3DDISTANCEUNIT }
• enum _spinScan3dCoordinateSystemEnums {
Scan3dCoordinateSystem_Cartesian,
Scan3dCoordinateSystem_Spherical,
Scan3dCoordinateSystem_Cylindrical,
NUM_SCAN3DCOORDINATESYSTEM }
• enum _spinScan3dOutputModeEnums {
Scan3dOutputMode_UncalibratedC,
Scan3dOutputMode_CalibratedABC_Grid,
Scan3dOutputMode_CalibratedABC_PointCloud,
Scan3dOutputMode_CalibratedAC,
Scan3dOutputMode_CalibratedAC_Linescan,
Scan3dOutputMode_CalibratedC,
Scan3dOutputMode_CalibratedC_Linescan,
Scan3dOutputMode_RectifiedC,

```

```

Scan3dOutputMode_RectifiedC_Linescan,
Scan3dOutputMode_DisparityC,
Scan3dOutputMode_DisparityC_Linescan,
NUM_SCAN3DOUTPUTMODE }

• enum _spinScan3dCoordinateSystemReferenceEnums {
Scan3dCoordinateSystemReference_Anchor,
Scan3dCoordinateSystemReference_Transformed,
NUM_SCAN3DCOORDINATESYSTEMREFERENCE }

• enum _spinScan3dCoordinateSelectorEnums {
Scan3dCoordinateSelector_CoordinateA,
Scan3dCoordinateSelector_CoordinateB,
Scan3dCoordinateSelector_CoordinateC,
NUM_SCAN3DCOORDINATESELECTOR }

• enum _spinScan3dCoordinateTransformSelectorEnums {
Scan3dCoordinateTransformSelector_RotationX,
Scan3dCoordinateTransformSelector_RotationY,
Scan3dCoordinateTransformSelector_RotationZ,
Scan3dCoordinateTransformSelector_TranslationX,
Scan3dCoordinateTransformSelector_TranslationY,
Scan3dCoordinateTransformSelector_TranslationZ,
NUM_SCAN3DCOORDINATETRANSFORMSELECTOR }

• enum _spinScan3dCoordinateReferenceSelectorEnums {
Scan3dCoordinateReferenceSelector_RotationX,
Scan3dCoordinateReferenceSelector_RotationY,
Scan3dCoordinateReferenceSelector_RotationZ,
Scan3dCoordinateReferenceSelector_TranslationX,
Scan3dCoordinateReferenceSelector_TranslationY,
Scan3dCoordinateReferenceSelector_TranslationZ,
NUM_SCAN3DCOORDINATEREFERENCESELECTOR }

• enum _spinChunkImageComponentEnums {
ChunkImageComponent_Intensity,
ChunkImageComponent_Color,
ChunkImageComponent_Infrared,
ChunkImageComponent_Ultraviolet,
ChunkImageComponent_Range,
ChunkImageComponent_Disparity,
ChunkImageComponent_Confidence,
ChunkImageComponent_Scatter,
NUM_CHUNKIMAGECOMPONENT }

• enum _spinChunkCounterSelectorEnums {
ChunkCounterSelector_Counter0,
ChunkCounterSelector_Counter1,
ChunkCounterSelector_Counter2,
NUM_CHUNKCOUNTERSELECTOR }

• enum _spinChunkTimerSelectorEnums {
ChunkTimerSelector_Timer0,
ChunkTimerSelector_Timer1,
ChunkTimerSelector_Timer2,
NUM_CHUNKTIMERSELECTOR }

• enum _spinChunkEncoderSelectorEnums {
ChunkEncoderSelector_Encoder0,
ChunkEncoderSelector_Encoder1,
ChunkEncoderSelector_Encoder2,
NUM_CHUNKENCODERSELECTOR }

• enum _spinChunkEncoderStatusEnums {
ChunkEncoderStatus_EncoderUp,
ChunkEncoderStatus_EncoderDown,
ChunkEncoderStatus_EncoderIdle,

```

- ```

ChunkEncoderStatus_EncoderStatic,
NUM_CHUNKENCODERSTATUS }

```
- `enum _spinChunkExposureTimeSelectorEnums {`  
`ChunkExposureTimeSelector_Common,`  
`ChunkExposureTimeSelector_Red,`  
`ChunkExposureTimeSelector_Green,`  
`ChunkExposureTimeSelector_Blue,`  
`ChunkExposureTimeSelector_Cyan,`  
`ChunkExposureTimeSelector_Magenta,`  
`ChunkExposureTimeSelector_Yellow,`  
`ChunkExposureTimeSelector_Infrared,`  
`ChunkExposureTimeSelector_Ultraviolet,`  
`ChunkExposureTimeSelector_Stage1,`  
`ChunkExposureTimeSelector_Stage2,`  
`NUM_CHUNKEXPOSURETIMESELECTOR }`
  - `enum _spinChunkSourceIDEnums {`  
`ChunkSourceID_Source0,`  
`ChunkSourceID_Source1,`  
`ChunkSourceID_Source2,`  
`NUM_CHUNKSOURCEID }`
  - `enum _spinChunkRegionIDEnums {`  
`ChunkRegionID_Region0,`  
`ChunkRegionID_Region1,`  
`ChunkRegionID_Region2,`  
`NUM_CHUNKREGIONID }`
  - `enum _spinChunkTransferStreamIDEnums {`  
`ChunkTransferStreamID_Stream0,`  
`ChunkTransferStreamID_Stream1,`  
`ChunkTransferStreamID_Stream2,`  
`ChunkTransferStreamID_Stream3,`  
`NUM_CHUNKTRANSFERSTREAMID }`
  - `enum _spinChunkScan3dDistanceUnitEnums {`  
`ChunkScan3dDistanceUnit_Millimeter,`  
`ChunkScan3dDistanceUnit_Inch,`  
`NUM_CHUNKSCAN3DDISTANCEUNIT }`
  - `enum _spinChunkScan3dOutputModeEnums {`  
`ChunkScan3dOutputMode_UncalibratedC,`  
`ChunkScan3dOutputMode_CalibratedABC_Grid,`  
`ChunkScan3dOutputMode_CalibratedABC_PointCloud,`  
`ChunkScan3dOutputMode_CalibratedAC,`  
`ChunkScan3dOutputMode_CalibratedAC_Linescan,`  
`ChunkScan3dOutputMode_CalibratedC,`  
`ChunkScan3dOutputMode_CalibratedC_Linescan,`  
`ChunkScan3dOutputMode_RectifiedC,`  
`ChunkScan3dOutputMode_RectifiedC_Linescan,`  
`ChunkScan3dOutputMode_DisparityC,`  
`ChunkScan3dOutputMode_DisparityC_Linescan,`  
`NUM_CHUNKSCAN3DOUTPUTMODE }`
  - `enum _spinChunkScan3dCoordinateSystemEnums {`  
`ChunkScan3dCoordinateSystem_Cartesian,`  
`ChunkScan3dCoordinateSystem_Spherical,`  
`ChunkScan3dCoordinateSystem_Cylindrical,`  
`NUM_CHUNKSCAN3DCOORDINATESYSTEM }`
  - `enum _spinChunkScan3dCoordinateSystemReferenceEnums {`  
`ChunkScan3dCoordinateSystemReference_Anchor,`  
`ChunkScan3dCoordinateSystemReference_Transformed,`  
`NUM_CHUNKSCAN3DCOORDINATESYSTEMREFERENCE }`

- `enum _spinChunkScan3dCoordinateSelectorEnums {`  
`ChunkScan3dCoordinateSelector_CoordinateA,`  
`ChunkScan3dCoordinateSelector_CoordinateB,`  
`ChunkScan3dCoordinateSelector_CoordinateC,`  
`NUM_CHUNKSCAN3DCOORDINATESELECTOR }`
- `enum _spinChunkScan3dCoordinateTransformSelectorEnums {`  
`ChunkScan3dCoordinateTransformSelector_RotationX,`  
`ChunkScan3dCoordinateTransformSelector_RotationY,`  
`ChunkScan3dCoordinateTransformSelector_RotationZ,`  
`ChunkScan3dCoordinateTransformSelector_TranslationX,`  
`ChunkScan3dCoordinateTransformSelector_TranslationY,`  
`ChunkScan3dCoordinateTransformSelector_TranslationZ,`  
`NUM_CHUNKSCAN3DCOORDINATETRANSFORMSELECTOR }`
- `enum _spinChunkScan3dCoordinateReferenceSelectorEnums {`  
`ChunkScan3dCoordinateReferenceSelector_RotationX,`  
`ChunkScan3dCoordinateReferenceSelector_RotationY,`  
`ChunkScan3dCoordinateReferenceSelector_RotationZ,`  
`ChunkScan3dCoordinateReferenceSelector_TranslationX,`  
`ChunkScan3dCoordinateReferenceSelector_TranslationY,`  
`ChunkScan3dCoordinateReferenceSelector_TranslationZ,`  
`NUM_CHUNKSCAN3DCOORDINATEREFERENCESELECTOR }`
- `enum _spinDeviceTapGeometryEnums {`  
`DeviceTapGeometry_Geometry_1X_1Y,`  
`DeviceTapGeometry_Geometry_1X2_1Y,`  
`DeviceTapGeometry_Geometry_1X2_1Y2,`  
`DeviceTapGeometry_Geometry_2X_1Y,`  
`DeviceTapGeometry_Geometry_2X_1Y2Geometry_2XE_1Y,`  
`DeviceTapGeometry_Geometry_2XE_1Y2,`  
`DeviceTapGeometry_Geometry_2XM_1Y,`  
`DeviceTapGeometry_Geometry_2XM_1Y2,`  
`DeviceTapGeometry_Geometry_1X_1Y2,`  
`DeviceTapGeometry_Geometry_1X_2YE,`  
`DeviceTapGeometry_Geometry_1X3_1Y,`  
`DeviceTapGeometry_Geometry_3X_1Y,`  
`DeviceTapGeometry_Geometry_1X,`  
`DeviceTapGeometry_Geometry_1X2,`  
`DeviceTapGeometry_Geometry_2X,`  
`DeviceTapGeometry_Geometry_2XE,`  
`DeviceTapGeometry_Geometry_2XM,`  
`DeviceTapGeometry_Geometry_1X3,`  
`DeviceTapGeometry_Geometry_3X,`  
`DeviceTapGeometry_Geometry_1X4_1Y,`  
`DeviceTapGeometry_Geometry_4X_1Y,`  
`DeviceTapGeometry_Geometry_2X2_1Y,`  
`DeviceTapGeometry_Geometry_2X2E_1YGeometry_2X2M_1Y,`  
`DeviceTapGeometry_Geometry_1X2_2YE,`  
`DeviceTapGeometry_Geometry_2X_2YE,`  
`DeviceTapGeometry_Geometry_2XE_2YE,`  
`DeviceTapGeometry_Geometry_2XM_2YE,`  
`DeviceTapGeometry_Geometry_1X4,`  
`DeviceTapGeometry_Geometry_4X,`  
`DeviceTapGeometry_Geometry_2X2,`  
`DeviceTapGeometry_Geometry_2X2E,`  
`DeviceTapGeometry_Geometry_2X2M,`  
`DeviceTapGeometry_Geometry_1X8_1Y,`  
`DeviceTapGeometry_Geometry_8X_1Y,`  
`DeviceTapGeometry_Geometry_4X2_1Y,`  
`DeviceTapGeometry_Geometry_2X2E_2YE,`



```

DeviceTapGeometry_Geometry_1X8,
DeviceTapGeometry_Geometry_8X,
DeviceTapGeometry_Geometry_4X2,
DeviceTapGeometry_Geometry_4X2E,
DeviceTapGeometry_Geometry_4X2E_1Y,
DeviceTapGeometry_Geometry_1X10_1Y,
DeviceTapGeometry_Geometry_10X_1Y,
DeviceTapGeometry_Geometry_1X10,
DeviceTapGeometry_Geometry_10X,
NUM_DEVICETAPGEOMETRY }

• enum _spinGevPhysicalLinkConfigurationEnums {
    GevPhysicalLinkConfiguration_SingleLink,
    GevPhysicalLinkConfiguration_MultiLink,
    GevPhysicalLinkConfiguration_StaticLAG,
    GevPhysicalLinkConfiguration_DynamicLAG,
    NUM_GEVPHYSICALLINKCONFIGURATION }

• enum _spinGevCurrentPhysicalLinkConfigurationEnums {
    GevCurrentPhysicalLinkConfiguration_SingleLink,
    GevCurrentPhysicalLinkConfiguration_MultiLink,
    GevCurrentPhysicalLinkConfiguration_StaticLAG,
    GevCurrentPhysicalLinkConfiguration_DynamicLAG,
    NUM_GEVCURRENTPHYSICALLINKCONFIGURATION }

• enum _spinGevIPConfigurationStatusEnums {
    GevIPConfigurationStatus_None,
    GevIPConfigurationStatus_PersistentIP,
    GevIPConfigurationStatus_DHCP,
    GevIPConfigurationStatus_LLA,
    GevIPConfigurationStatus_ForceIP,
    NUM_GEVIPCONFIGURATIONSTATUS }

• enum _spinGevGVCPExtendedStatusCodesSelectorEnums {
    GevGVCPExtendedStatusCodesSelector_Version1_1,
    GevGVCPExtendedStatusCodesSelector_Version2_0,
    NUM_GEVGVCPEXTENDEDSTATUSCODESSELECTOR }

• enum _spinGevGVSPExtendedIDModeEnums {
    GevGVSPExtendedIDMode_Off,
    GevGVSPExtendedIDMode_On,
    NUM_GEVGVSPEXTENDEDIDMODE }

• enum _spinCIConfigurationEnums {
    CIConfiguration_Base,
    CIConfiguration_Medium,
    CIConfiguration_Full,
    CIConfiguration_DualBase,
    CIConfiguration_EightyBit,
    NUM_CLCONFIGURATION }

• enum _spinCITimeSlotsCountEnums {
    CITimeSlotsCount_One,
    CITimeSlotsCount_Two,
    CITimeSlotsCount_Three,
    NUM_CLTIMESLOTSCOUNT }

• enum _spinCxpLinkConfigurationStatusEnums {
    CxpLinkConfigurationStatus_None,
    CxpLinkConfigurationStatus_Pending,
    CxpLinkConfigurationStatus_CXP1_X1,
    CxpLinkConfigurationStatus_CXP2_X1,
    CxpLinkConfigurationStatus_CXP3_X1,
    CxpLinkConfigurationStatus_CXP5_X1,
    CxpLinkConfigurationStatus_CXP6_X1,
    CxpLinkConfigurationStatus_CXP1_X2,

```

```

CxpLinkConfigurationStatus_CXP2_X2,
CxpLinkConfigurationStatus_CXP3_X2,
CxpLinkConfigurationStatus_CXP5_X2,
CxpLinkConfigurationStatus_CXP6_X2,
CxpLinkConfigurationStatus_CXP1_X3,
CxpLinkConfigurationStatus_CXP2_X3,
CxpLinkConfigurationStatus_CXP3_X3,
CxpLinkConfigurationStatus_CXP5_X3,
CxpLinkConfigurationStatus_CXP6_X3,
CxpLinkConfigurationStatus_CXP1_X4,
CxpLinkConfigurationStatus_CXP2_X4,
CxpLinkConfigurationStatus_CXP3_X4,
CxpLinkConfigurationStatus_CXP5_X4,
CxpLinkConfigurationStatus_CXP6_X4,
CxpLinkConfigurationStatus_CXP1_X5,
CxpLinkConfigurationStatus_CXP2_X5,
CxpLinkConfigurationStatus_CXP3_X5,
CxpLinkConfigurationStatus_CXP5_X5,
CxpLinkConfigurationStatus_CXP6_X5,
CxpLinkConfigurationStatus_CXP1_X6,
CxpLinkConfigurationStatus_CXP2_X6,
CxpLinkConfigurationStatus_CXP3_X6,
CxpLinkConfigurationStatus_CXP5_X6,
CxpLinkConfigurationStatus_CXP6_X6,
NUM_CXPLINKCONFIGURATIONSTATUS }

• enum _spinCxpLinkConfigurationPreferredEnums {
CxpLinkConfigurationPreferred_CXP1_X1,
CxpLinkConfigurationPreferred_CXP2_X1,
CxpLinkConfigurationPreferred_CXP3_X1,
CxpLinkConfigurationPreferred_CXP5_X1,
CxpLinkConfigurationPreferred_CXP6_X1,
CxpLinkConfigurationPreferred_CXP1_X2,
CxpLinkConfigurationPreferred_CXP2_X2,
CxpLinkConfigurationPreferred_CXP3_X2,
CxpLinkConfigurationPreferred_CXP5_X2,
CxpLinkConfigurationPreferred_CXP6_X2,
CxpLinkConfigurationPreferred_CXP1_X3,
CxpLinkConfigurationPreferred_CXP2_X3,
CxpLinkConfigurationPreferred_CXP3_X3,
CxpLinkConfigurationPreferred_CXP5_X3,
CxpLinkConfigurationPreferred_CXP6_X3,
CxpLinkConfigurationPreferred_CXP1_X4,
CxpLinkConfigurationPreferred_CXP2_X4,
CxpLinkConfigurationPreferred_CXP3_X4,
CxpLinkConfigurationPreferred_CXP5_X4,
CxpLinkConfigurationPreferred_CXP6_X4,
CxpLinkConfigurationPreferred_CXP1_X5,
CxpLinkConfigurationPreferred_CXP2_X5,
CxpLinkConfigurationPreferred_CXP3_X5,
CxpLinkConfigurationPreferred_CXP5_X5,
CxpLinkConfigurationPreferred_CXP6_X5,
CxpLinkConfigurationPreferred_CXP1_X6,
CxpLinkConfigurationPreferred_CXP2_X6,
CxpLinkConfigurationPreferred_CXP3_X6,
CxpLinkConfigurationPreferred_CXP5_X6,
CxpLinkConfigurationPreferred_CXP6_X6,
NUM_CXPLINKCONFIGURATIONPREFERRED }

• enum _spinCxpLinkConfigurationEnums {

```

```

CxpLinkConfiguration_Auto,
CxpLinkConfiguration_CXP1_X1,
CxpLinkConfiguration_CXP2_X1,
CxpLinkConfiguration_CXP3_X1,
CxpLinkConfiguration_CXP5_X1,
CxpLinkConfiguration_CXP6_X1,
CxpLinkConfiguration_CXP1_X2,
CxpLinkConfiguration_CXP2_X2,
CxpLinkConfiguration_CXP3_X2,
CxpLinkConfiguration_CXP5_X2,
CxpLinkConfiguration_CXP6_X2,
CxpLinkConfiguration_CXP1_X3,
CxpLinkConfiguration_CXP2_X3,
CxpLinkConfiguration_CXP3_X3,
CxpLinkConfiguration_CXP5_X3,
CxpLinkConfiguration_CXP6_X3,
CxpLinkConfiguration_CXP1_X4,
CxpLinkConfiguration_CXP2_X4,
CxpLinkConfiguration_CXP3_X4,
CxpLinkConfiguration_CXP5_X4,
CxpLinkConfiguration_CXP6_X4,
CxpLinkConfiguration_CXP1_X5,
CxpLinkConfiguration_CXP2_X5,
CxpLinkConfiguration_CXP3_X5,
CxpLinkConfiguration_CXP5_X5,
CxpLinkConfiguration_CXP6_X5,
CxpLinkConfiguration_CXP1_X6,
CxpLinkConfiguration_CXP2_X6,
CxpLinkConfiguration_CXP3_X6,
CxpLinkConfiguration_CXP5_X6,
CxpLinkConfiguration_CXP6_X6,
NUM_CXPLINKCONFIGURATION }
• enum _spinCxpConnectionTestModeEnums {
  CxpConnectionTestMode_Off,
  CxpConnectionTestMode_Mode1,
  NUM_CXPCONNECTIONTESTMODE }
• enum _spinCxpPoCxpStatusEnums {
  CxpPoCxpStatus_Auto,
  CxpPoCxpStatus_Off,
  CxpPoCxpStatus_Tripped,
  NUM_CXPPOCXPSTATUS }

```

## 6.1.1 Enumeration Type Documentation

### 6.1.1.1 \_spinAcquisitionModeEnums

```
enum _spinAcquisitionModeEnums
```

< Sets the acquisition mode of the device. Continuous: acquires images continuously. Multi Frame: acquires a specified number of images before stopping acquisition. Single Frame: acquires 1 image before stopping acquisition.

**Enumerator**

|                             |  |
|-----------------------------|--|
| AcquisitionMode_Continuous  |  |
| AcquisitionMode_SingleFrame |  |
| AcquisitionMode_MultiFrame  |  |
| NUM_ACQUISITIONMODE         |  |

**6.1.1.2 \_spinAcquisitionStatusSelectorEnums**

```
enum _spinAcquisitionStatusSelectorEnums
```

< Selects the internal acquisition signal to read using AcquisitionStatus.

**Enumerator**

|                                                  |                                                                                  |
|--------------------------------------------------|----------------------------------------------------------------------------------|
| AcquisitionStatusSelector_AcquisitionTriggerWait | Device is currently waiting for a trigger for the capture of one or many frames. |
| AcquisitionStatusSelector_AcquisitionActive      | Device is currently doing an acquisition of one or many frames.                  |
| AcquisitionStatusSelector_AcquisitionTransfer    | Device is currently transferring an acquisition of one or many frames.           |
| AcquisitionStatusSelector_FrameTriggerWait       | Device is currently waiting for a frame start trigger.                           |
| AcquisitionStatusSelector_FrameActive            | Device is currently doing the capture of a frame.                                |
| AcquisitionStatusSelector_ExposureActive         | Device is doing the exposure of a frame.                                         |
| NUM_ACQUISITIONSTATUSSELECTION                   |                                                                                  |

**6.1.1.3 \_spinActionUnconditionalModeEnums**

```
enum _spinActionUnconditionalModeEnums
```

< Enables the unconditional action command mode where action commands are processed even when the primary control channel is closed.

**Enumerator**

|                             |                                 |
|-----------------------------|---------------------------------|
| ActionUnconditionalMode_Off | Unconditional mode is disabled. |
| ActionUnconditionalMode_On  | Unconditional mode is enabled.  |
| NUM_ACTIONUNCONDITIONALMODE |                                 |

**6.1.1.4 \_spinAdcBitDepthEnums**

```
enum _spinAdcBitDepthEnums
```

< Selects which ADC bit depth to use. A higher ADC bit depth results in better image quality but slower maximum frame rate.

#### Enumerator

|                   |  |
|-------------------|--|
| AdcBitDepth_Bit8  |  |
| AdcBitDepth_Bit10 |  |
| AdcBitDepth_Bit12 |  |
| AdcBitDepth_Bit14 |  |
| NUM_ADCBITDEPTH   |  |

#### 6.1.1.5 \_spinAutoAlgorithmSelectorEnums

```
enum _spinAutoAlgorithmSelectorEnums
```

< Selects which Auto Algorithm is controlled by the RoiEnable, OffsetX, OffsetY, Width, Height features.

#### Enumerator

|                           |                                           |
|---------------------------|-------------------------------------------|
| AutoAlgorithmSelector_Awb | Selects the Auto White Balance algorithm. |
| AutoAlgorithmSelector_Ae  | Selects the Auto Exposure algorithm.      |
| NUM_AUTOALGORITHMSELECTOR |                                           |

#### 6.1.1.6 \_spinAutoExposureControlPriorityEnums

```
enum _spinAutoExposureControlPriorityEnums
```

< Selects whether to adjust gain or exposure first. When gain priority is selected, the camera fixes the gain to 0 dB, and the exposure is adjusted according to the target grey level. If the maximum exposure is reached before the target grey level is hit, the gain starts to change to meet the target. This mode is used to have the minimum noise. When exposure priority is selected, the camera sets the exposure to a small value (default is 5 ms). The gain is adjusted according to the target grey level. If maximum gain is reached before the target grey level is hit, the exposure starts to change to meet the target. This mode is used to capture fast motion.

#### Enumerator

|                                          |  |
|------------------------------------------|--|
| AutoExposureControlPriority_Gain         |  |
| AutoExposureControlPriority_ExposureTime |  |
| NUM_AUTOEXPOSURECONTROLPRIORITY          |  |

#### 6.1.1.7 \_spinAutoExposureLightingModeEnums

```
enum _spinAutoExposureLightingModeEnums
```

< Selects a lighting mode: Backlight, Frontlight or Normal (default). a. Backlight compensation: used when a strong light is coming from the back of the object. b. Frontlight compensation: used when a strong light is shining in the front of the object while the background is dark. c. Normal lighting: used when the object is not under backlight or frontlight conditions. When normal lighting is selected, metering modes are available.

#### Enumerator

|                                     |  |
|-------------------------------------|--|
| AutoExposureLightingMode_AutoDetect |  |
| AutoExposureLightingMode_Backlight  |  |
| AutoExposureLightingMode_Frontlight |  |
| AutoExposureLightingMode_Normal     |  |
| NUM_AUTOEXPOSURELIGHTINGMODE        |  |

### 6.1.1.8 \_spinAutoExposureMeteringModeEnums

```
enum _spinAutoExposureMeteringModeEnums
```

< Selects a metering mode: average, spot, or partial metering. a. Average: Measures the light from the entire scene uniformly to determine the final exposure value. Every portion of the exposed area has the same contribution. b. Spot: Measures a small area (about 3%) in the center of the scene while the rest of the scene is ignored. This mode is used when the scene has a high contrast and the object of interest is relatively small. c. Partial: Measures the light from a larger area (about 11%) in the center of the scene. This mode is used when very dark or bright regions appear at the edge of the frame. Note: Metering mode is available only when Lighting Mode Selector is Normal.

#### Enumerator

|                                         |  |
|-----------------------------------------|--|
| AutoExposureMeteringMode_Average        |  |
| AutoExposureMeteringMode_Spot           |  |
| AutoExposureMeteringMode_Partial        |  |
| AutoExposureMeteringMode_CenterWeighted |  |
| AutoExposureMeteringMode_HistogramPeak  |  |
| NUM_AUTOEXPOSUREMETERINGMODE            |  |

### 6.1.1.9 \_spinAutoExposureTargetGreyValueAutoEnums

```
enum _spinAutoExposureTargetGreyValueAutoEnums
```

< This indicates whether the target image grey level is automatically set by the camera or manually set by the user. Note that the target grey level is in the linear domain before gamma correction is applied.

#### Enumerator

|                                            |                                                                                      |
|--------------------------------------------|--------------------------------------------------------------------------------------|
| AutoExposureTargetGreyValueAuto_Off        | Target grey value is manually controlled                                             |
| AutoExposureTargetGreyValueAuto_Continuous | Target grey value is constantly adapted by the device to maximize the dynamic range. |
| NUM_AUTOEXPOSURETARGETGREYVALUEAUTO        |                                                                                      |

#### 6.1.1.10 \_spinBalanceRatioSelectorEnums

enum `_spinBalanceRatioSelectorEnums`

< Selects a balance ratio to configure once a balance ratio control has been selected.

##### Enumerator

|                           |                                                                                                                 |
|---------------------------|-----------------------------------------------------------------------------------------------------------------|
| BalanceRatioSelector_Red  | Selects the red balance ratio control for adjustment. The red balance ratio is relative to the green channel.   |
| BalanceRatioSelector_Blue | Selects the blue balance ratio control for adjustment. The blue balance ratio is relative to the green channel. |
| NUM_BALANCERATIOSELECTOR  |                                                                                                                 |

#### 6.1.1.11 \_spinBalanceWhiteAutoEnums

enum `_spinBalanceWhiteAutoEnums`

< White Balance compensates for color shifts caused by different lighting conditions. It can be automatically or manually controlled. For manual control, set to Off. For automatic control, set to Once or Continuous.

##### Enumerator

|                             |                                                                                                            |
|-----------------------------|------------------------------------------------------------------------------------------------------------|
| BalanceWhiteAuto_Off        | Sets operation mode to Off, which is manual control.                                                       |
| BalanceWhiteAuto_Once       | Sets operation mode to once. Once runs for a number of iterations and then sets White Balance Auto to Off. |
| BalanceWhiteAuto_Continuous | Sets operation mode to continuous. Continuous automatically adjusts values if the colors are imbalanced.   |
| NUM_BALANCEWHITEAUTO        |                                                                                                            |

#### 6.1.1.12 \_spinBalanceWhiteAutoProfileEnums

enum `_spinBalanceWhiteAutoProfileEnums`

< Selects the profile used by BalanceWhiteAuto.

##### Enumerator

|                                 |                                                                                       |
|---------------------------------|---------------------------------------------------------------------------------------|
| BalanceWhiteAutoProfile_Indoor  | Indoor auto white balance Profile. Can be used to compensate for artificial lighting. |
| BalanceWhiteAutoProfile_Outdoor | Outdoor auto white balance profile. Designed for scenes with natural lighting.        |
| NUM_BALANCEWHITEAUTOPROFILE     |                                                                                       |

### 6.1.1.13 \_spinBinningHorizontalModeEnums

```
enum _spinBinningHorizontalModeEnums
```

<

#### Enumerator

|                               |                                                                                                                                                  |
|-------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| BinningHorizontalMode_Sum     | The response from the combined horizontal cells is added, resulting in increased sensitivity (a brighter image).                                 |
| BinningHorizontalMode_Average | The response from the combined horizontal cells is averaged, resulting in increased signal/noise ratio. Not all sensors support average binning. |
| NUM_BINNINGHORIZONTALMODE     |                                                                                                                                                  |

### 6.1.1.14 \_spinBinningSelectorEnums

```
enum _spinBinningSelectorEnums
```

< Selects which binning engine is controlled by the BinningHorizontal and BinningVertical features.

#### Enumerator

|                        |                                                                                                                                          |
|------------------------|------------------------------------------------------------------------------------------------------------------------------------------|
| BinningSelector_All    | The total amount of binning to be performed on the captured sensor data.                                                                 |
| BinningSelector_Sensor | The portion of binning to be performed on the sensor directly.                                                                           |
| BinningSelector_ISP    | The portion of binning to be performed by the image signal processing engine (ISP) outside of the sensor. Note: the ISP can be disabled. |
| NUM_BINNINGSELECTOR    |                                                                                                                                          |

### 6.1.1.15 \_spinBinningVerticalModeEnums

```
enum _spinBinningVerticalModeEnums
```

<

#### Enumerator

|                             |                                                                                                                                                |
|-----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------|
| BinningVerticalMode_Sum     | The response from the combined vertical cells is added, resulting in increased sensitivity (a brighter image).                                 |
| BinningVerticalMode_Average | The response from the combined vertical cells is averaged, resulting in increased signal/noise ratio. Not all sensors support average binning. |
| NUM_BINNINGVERTICALMODE     |                                                                                                                                                |



**6.1.1.16 \_spinBlackLevelAutoBalanceEnums**

```
enum _spinBlackLevelAutoBalanceEnums
```

< Controls the mode for automatic black level balancing between the sensor color channels or taps. The black level coefficients of each channel are adjusted so they are matched.

**Enumerator**

|                                  |                                                                                                                                           |
|----------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------|
| BlackLevelAutoBalance_Off        | Black level tap balancing is user controlled using BlackLevel.                                                                            |
| BlackLevelAutoBalance_Once       | Black level tap balancing is automatically adjusted once by the device. Once it has converged, it automatically returns to the Off state. |
| BlackLevelAutoBalance_Continuous | Black level tap balancing is constantly adjusted by the device.                                                                           |
| NUM_BLACKLEVELAUTOBALANCE        |                                                                                                                                           |

**6.1.1.17 \_spinBlackLevelAutoEnums**

```
enum _spinBlackLevelAutoEnums
```

< Controls the mode for automatic black level adjustment. The exact algorithm used to implement this adjustment is device-specific.

**Enumerator**

|                           |                                                                                                                                    |
|---------------------------|------------------------------------------------------------------------------------------------------------------------------------|
| BlackLevelAuto_Off        | Analog black level is user controlled using BlackLevel.                                                                            |
| BlackLevelAuto_Once       | Analog black level is automatically adjusted once by the device. Once it has converged, it automatically returns to the Off state. |
| BlackLevelAuto_Continuous | Analog black level is constantly adjusted by the device.                                                                           |
| NUM_BLACKLEVELAUTO        |                                                                                                                                    |

**6.1.1.18 \_spinBlackLevelSelectorEnums**

```
enum _spinBlackLevelSelectorEnums
```

< Selects which black level to control. Only All can be set by the user. Analog and Digital are read-only.

**Enumerator**

|                            |  |
|----------------------------|--|
| BlackLevelSelector_All     |  |
| BlackLevelSelector_Analog  |  |
| BlackLevelSelector_Digital |  |
| NUM_BLACKLEVELSELECTOR     |  |

### 6.1.1.19 \_spinChunkBlackLevelSelectorEnums

```
enum _spinChunkBlackLevelSelectorEnums
```

< Selects which black level to retrieve

Enumerator

|                             |  |
|-----------------------------|--|
| ChunkBlackLevelSelector_All |  |
| NUM_CHUNKBLACKLEVELSELECTOR |  |

### 6.1.1.20 \_spinChunkCounterSelectorEnums

```
enum _spinChunkCounterSelectorEnums
```

< Selects which counter to retrieve data from.

Enumerator

|                               |                        |
|-------------------------------|------------------------|
| ChunkCounterSelector_Counter0 | Selects the counter 0. |
| ChunkCounterSelector_Counter1 | Selects the counter 1. |
| ChunkCounterSelector_Counter2 | Selects the counter 2. |
| NUM_CHUNKCOUNTERSELECTOR      |                        |

### 6.1.1.21 \_spinChunkEncoderSelectorEnums

```
enum _spinChunkEncoderSelectorEnums
```

< Selects which Encoder to retrieve data from.

Enumerator

|                               |                             |
|-------------------------------|-----------------------------|
| ChunkEncoderSelector_Encoder0 | Selects the first Encoder.  |
| ChunkEncoderSelector_Encoder1 | Selects the first Encoder.  |
| ChunkEncoderSelector_Encoder2 | Selects the second Encoder. |
| NUM_CHUNKENCODERSELECTOR      |                             |

### 6.1.1.22 \_spinChunkEncoderStatusEnums

```
enum _spinChunkEncoderStatusEnums
```

< Returns the motion status of the selected encoder.

#### Enumerator

|                                  |                                           |
|----------------------------------|-------------------------------------------|
| ChunkEncoderStatus_EncoderUp     | The encoder counter last incremented.     |
| ChunkEncoderStatus_EncoderDown   | The encoder counter last decremented.     |
| ChunkEncoderStatus_EncoderIdle   | The encoder is not active.                |
| ChunkEncoderStatus_EncoderStatic | No motion within the EncoderTimeout time. |
| NUM_CHUNKENCODERSTATUS           |                                           |

#### 6.1.1.23 \_spinChunkExposureTimeSelectorEnums

```
enum _spinChunkExposureTimeSelectorEnums
```

< Selects which exposure time is read by the ChunkExposureTime feature.

#### Enumerator

|                                       |                                        |
|---------------------------------------|----------------------------------------|
| ChunkExposureTimeSelector_Common      | Selects the common ExposureTime.       |
| ChunkExposureTimeSelector_Red         | Selects the red common ExposureTime.   |
| ChunkExposureTimeSelector_Green       | Selects the green ExposureTime.        |
| ChunkExposureTimeSelector_Blue        | Selects the blue ExposureTime.         |
| ChunkExposureTimeSelector_Cyan        | Selects the cyan common ExposureTime.. |
| ChunkExposureTimeSelector_Magenta     | Selects the magenta ExposureTime..     |
| ChunkExposureTimeSelector_Yellow      | Selects the yellow ExposureTime..      |
| ChunkExposureTimeSelector_Infrared    | Selects the infrared ExposureTime.     |
| ChunkExposureTimeSelector_Ultraviolet | Selects the ultraviolet ExposureTime.  |
| ChunkExposureTimeSelector_Stage1      | Selects the first stage ExposureTime.  |
| ChunkExposureTimeSelector_Stage2      | Selects the second stage ExposureTime. |
| NUM_CHUNKEXPOSURETIMESELECTOR         |                                        |

#### 6.1.1.24 \_spinChunkGainSelectorEnums

```
enum _spinChunkGainSelectorEnums
```

< Selects which gain to retrieve

#### Enumerator

|                         |  |
|-------------------------|--|
| ChunkGainSelector_All   |  |
| ChunkGainSelector_Red   |  |
| ChunkGainSelector_Green |  |
| ChunkGainSelector_Blue  |  |
| NUM_CHUNKGAINSELECTOR   |  |

### 6.1.1.25 `_spinChunkImageComponentEnums`

enum `_spinChunkImageComponentEnums`

< Returns the component of the payload image. This can be used to identify the image component of a generic part in a multipart transfer.

#### Enumerator

|                                              |                                                   |
|----------------------------------------------|---------------------------------------------------|
| <code>ChunkImageComponent_Intensity</code>   | The image data is the intensity component.        |
| <code>ChunkImageComponent_Color</code>       | The image data is color component.                |
| <code>ChunkImageComponent_Infrared</code>    | The image data is infrared component.             |
| <code>ChunkImageComponent_Ultraviolet</code> | The image data is the ultraviolet component.      |
| <code>ChunkImageComponent_Range</code>       | The image data is the range (distance) component. |
| <code>ChunkImageComponent_Disparity</code>   | The image data is the disparity component.        |
| <code>ChunkImageComponent_Confidence</code>  | The image data is the confidence map component.   |
| <code>ChunkImageComponent_Scatter</code>     | The image data is the scatter component.          |
| <code>NUM_CHUNKIMAGECOMPONENT</code>         |                                                   |

### 6.1.1.26 `_spinChunkPixelFormatEnums`

enum `_spinChunkPixelFormatEnums`

< Format of the pixel provided by the camera

#### Enumerator

|                                                     |  |
|-----------------------------------------------------|--|
| <code>ChunkPixelFormat_Mono8</code>                 |  |
| <code>ChunkPixelFormat_Mono12Packed</code>          |  |
| <code>ChunkPixelFormat_Mono16</code>                |  |
| <code>ChunkPixelFormat_RGB8Packed</code>            |  |
| <code>ChunkPixelFormat_YUV422Packed</code>          |  |
| <code>ChunkPixelFormat_BayerGR8</code>              |  |
| <code>ChunkPixelFormat_BayerRG8</code>              |  |
| <code>ChunkPixelFormat_BayerGB8</code>              |  |
| <code>ChunkPixelFormat_BayerBG8</code>              |  |
| <code>ChunkPixelFormat_YCbCr601_422_8_CbYCrY</code> |  |
| <code>NUM_CHUNKPIXELFORMAT</code>                   |  |

### 6.1.1.27 `_spinChunkRegionIDEnums`

enum `_spinChunkRegionIDEnums`

< Returns the identifier of Region that the image comes from.

#### Enumerator

|                       |                                |
|-----------------------|--------------------------------|
| ChunkRegionID_Region0 | Image comes from the Region 0. |
| ChunkRegionID_Region1 | Image comes from the Region 1. |
| ChunkRegionID_Region2 | Image comes from the Region 2. |
| NUM_CHUNKREGIONID     |                                |

#### 6.1.1.28 \_spinChunkScan3dCoordinateReferenceSelectorEnums

```
enum _spinChunkScan3dCoordinateReferenceSelectorEnums
```

< Selector to read a coordinate system reference value defining the transform of a point from one system to the other.

#### Enumerator

|                                                     |                         |
|-----------------------------------------------------|-------------------------|
| ChunkScan3dCoordinateReferenceSelector_RotationX    | Rotation around X axis. |
| ChunkScan3dCoordinateReferenceSelector_RotationY    | Rotation around Y axis. |
| ChunkScan3dCoordinateReferenceSelector_RotationZ    | Rotation around Z axis. |
| ChunkScan3dCoordinateReferenceSelector_TranslationX | X axis translation.     |
| ChunkScan3dCoordinateReferenceSelector_TranslationY | Y axis translation.     |
| ChunkScan3dCoordinateReferenceSelector_TranslationZ | Z axis translation.     |
| NUM_CHUNKSCAN3DCOORDINATEREFERENCESELECTOR          |                         |

#### 6.1.1.29 \_spinChunkScan3dCoordinateSelectorEnums

```
enum _spinChunkScan3dCoordinateSelectorEnums
```

< Selects which Coordinate to retrieve data from.

#### Enumerator

|                                           |                                   |
|-------------------------------------------|-----------------------------------|
| ChunkScan3dCoordinateSelector_CoordinateA | The first (X or Theta) coordinate |
| ChunkScan3dCoordinateSelector_CoordinateB | The second (Y or Phi) coordinate  |
| ChunkScan3dCoordinateSelector_CoordinateC | The third (Z or Rho) coordinate.  |
| NUM_CHUNKSCAN3DCOORDINATESELECTOR         |                                   |

#### 6.1.1.30 \_spinChunkScan3dCoordinateSystemEnums

```
enum _spinChunkScan3dCoordinateSystemEnums
```

< Returns the Coordinate System of the image included in the payload.

#### Enumerator

|                                         |                                                     |
|-----------------------------------------|-----------------------------------------------------|
| ChunkScan3dCoordinateSystem_Cartesian   | Default value. 3-axis orthogonal, right-hand X-Y-Z. |
| ChunkScan3dCoordinateSystem_Spherical   | A Theta-Phi-Rho coordinate system.                  |
| ChunkScan3dCoordinateSystem_Cylindrical | A Theta-Y-Rho coordinate system.                    |
| NUM_CHUNKSCAN3DCOORDINATESYSTEM         |                                                     |

#### 6.1.1.31 \_spinChunkScan3dCoordinateSystemReferenceEnums

enum `_spinChunkScan3dCoordinateSystemReferenceEnums`

< Returns the Coordinate System Position of the image included in the payload.

#### Enumerator

|                                                       |                                                                                                                                               |
|-------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| ChunkScan3dCoordinateSystemReference_Anchor           | Default value. Original fixed reference. The coordinate system fixed relative the camera reference point marker is used.                      |
| ChunkScan3dCoordinateSystemReference_↔<br>Transformed | Transformed reference system. The transformed coordinate system is used according to the definition in the rotation and translation matrices. |
| NUM_CHUNKSCAN3DCOORDINATESYSTEMRE↔<br>FERENCE         |                                                                                                                                               |

#### 6.1.1.32 \_spinChunkScan3dCoordinateTransformSelectorEnums

enum `_spinChunkScan3dCoordinateTransformSelectorEnums`

< Selector for transform values.

#### Enumerator

|                                                     |                           |
|-----------------------------------------------------|---------------------------|
| ChunkScan3dCoordinateTransformSelector_RotationX    | Rotation around X axis.   |
| ChunkScan3dCoordinateTransformSelector_RotationY    | Rotation around Y axis.   |
| ChunkScan3dCoordinateTransformSelector_RotationZ    | Rotation around Z axis.   |
| ChunkScan3dCoordinateTransformSelector_TranslationX | Translation along X axis. |
| ChunkScan3dCoordinateTransformSelector_TranslationY | Translation along Y axis. |
| ChunkScan3dCoordinateTransformSelector_TranslationZ | Translation along Z axis. |
| NUM_CHUNKSCAN3DCOORDINATETRANSFORMSELECTOR          |                           |

**6.1.1.33 \_spinChunkScan3dDistanceUnitEnums**

```
enum _spinChunkScan3dDistanceUnitEnums
```

< Returns the Distance Unit of the payload image.

**Enumerator**

|                                    |                                                         |
|------------------------------------|---------------------------------------------------------|
| ChunkScan3dDistanceUnit_Millimeter | Default value. Distance values are in millimeter units. |
| ChunkScan3dDistanceUnit_Inch       | Distance values are in inch units.                      |
| NUM_CHUNKSCAN3DDISTANCEUNIT        |                                                         |

**6.1.1.34 \_spinChunkScan3dOutputModeEnums**

```
enum _spinChunkScan3dOutputModeEnums
```

< Returns the Calibrated Mode of the payload image.

**Enumerator**

|                                                     |                                                                                                                                                                                                                                                                                                             |
|-----------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ChunkScan3dOutputMode_UncalibratedC                 | Uncalibrated 2.5D Depth map. The distance data does not represent a physical unit and may be non-linear. The data is a 2.5D range map only.                                                                                                                                                                 |
| ChunkScan3dOutputMode_CalibratedABC_Grid            | 3 Coordinates in grid organization. The full 3 coordinate data with the grid array organization from the sensor kept.                                                                                                                                                                                       |
| ChunkScan3dOutputMode_CalibratedABC_Point↔<br>Cloud | 3 Coordinates without organization. The full 3 coordinate data without any organization of data points. Typically only valid points transmitted giving varying image size.                                                                                                                                  |
| ChunkScan3dOutputMode_CalibratedAC                  | 2 Coordinates with fixed B sampling. The data is sent as a A and C coordinates (X,Z or Theta,Rho). The B (Y) axis uses the scale and offset parameters for the B axis.                                                                                                                                      |
| ChunkScan3dOutputMode_CalibratedAC_Linescan         | 2 Coordinates with varying sampling. The data is sent as a A and C coordinates (X,Z or Theta,Rho). The B (Y) axis comes from the encoder chunk value.                                                                                                                                                       |
| ChunkScan3dOutputMode_CalibratedC                   | Calibrated 2.5D Depth map. The distance data is expressed in the chosen distance unit. The data is a 2.5D range map. No information on X-Y axes available.                                                                                                                                                  |
| ChunkScan3dOutputMode_CalibratedC_Linescan          | Depth Map with varying B sampling. The distance data is expressed in the chosen distance unit. The data is a 2.5D range map. The B (Y) axis comes from the encoder chunk value.                                                                                                                             |
| ChunkScan3dOutputMode_RectifiedC                    | Rectified 2.5D Depth map. The distance data has been rectified to a uniform sampling pattern in the X and Y direction. The data is a 2.5D range map only. If a complete 3D point cloud is rectified but transmitted as explicit coordinates it should be transmitted as one of the "CalibratedABC" formats. |

## Enumerator

|                                           |                                                                                                                                                                             |
|-------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ChunkScan3dOutputMode_RectifiedC_Linescan | Rectified 2.5D Depth map with varying B sampling. The data is sent as rectified 1D profiles using Coord3D_C pixels. The B (Y) axis comes from the encoder chunk value.      |
| ChunkScan3dOutputMode_DisparityC          | Disparity 2.5D Depth map. The distance is inversely proportional to the pixel (disparity) value.                                                                            |
| ChunkScan3dOutputMode_DisparityC_Linescan | Disparity 2.5D Depth map with varying B sampling. The distance is inversely proportional to the pixel (disparity) value. The B (Y) axis comes from the encoder chunk value. |
| NUM_CHUNKSCAN3DOUTPUTMODE                 |                                                                                                                                                                             |

6.1.1.35 `_spinChunkSelectorEnums`

```
enum _spinChunkSelectorEnums
```

< Selects which chunk data to enable or disable.

## Enumerator

|                                        |  |
|----------------------------------------|--|
| ChunkSelector_Image                    |  |
| ChunkSelector_CRC                      |  |
| ChunkSelector_FrameID                  |  |
| ChunkSelector_OffsetX                  |  |
| ChunkSelector_OffsetY                  |  |
| ChunkSelector_Width                    |  |
| ChunkSelector_Height                   |  |
| ChunkSelector_ExposureTime             |  |
| ChunkSelector_Gain                     |  |
| ChunkSelector_BlackLevel               |  |
| ChunkSelector_PixelFormat              |  |
| ChunkSelector_Timestamp                |  |
| ChunkSelector_SequencerSetActive       |  |
| ChunkSelector_SerialData               |  |
| ChunkSelector_ExposureEndLineStatusAll |  |
| NUM_CHUNKSELECTOR                      |  |

6.1.1.36 `_spinChunkSourceIDEnums`

```
enum _spinChunkSourceIDEnums
```

< Returns the identifier of Source that the image comes from.



## Enumerator

|                       |                                |
|-----------------------|--------------------------------|
| ChunkSourceID_Source0 | Image comes from the Source 0. |
| ChunkSourceID_Source1 | Image comes from the Source 1. |
| ChunkSourceID_Source2 | Image comes from the Source 2. |
| NUM_CHUNKSOURCEID     |                                |

## 6.1.1.37 \_\_spinChunkTimerSelectorEnums

```
enum __spinChunkTimerSelectorEnums
```

< Selects which Timer to retrieve data from.

## Enumerator

|                           |                           |
|---------------------------|---------------------------|
| ChunkTimerSelector_Timer0 | Selects the first Timer.  |
| ChunkTimerSelector_Timer1 | Selects the first Timer.  |
| ChunkTimerSelector_Timer2 | Selects the second Timer. |
| NUM_CHUNKTIMERSELECTOR    |                           |

## 6.1.1.38 \_\_spinChunkTransferStreamIDEnums

```
enum __spinChunkTransferStreamIDEnums
```

< Returns identifier of the stream that generated this block.

## Enumerator

|                               |                          |
|-------------------------------|--------------------------|
| ChunkTransferStreamID_Stream0 | Data comes from Stream0. |
| ChunkTransferStreamID_Stream1 | Data comes from Stream1. |
| ChunkTransferStreamID_Stream2 | Data comes from Stream2. |
| ChunkTransferStreamID_Stream3 | Data comes from Stream3. |
| NUM_CHUNKTRANSFERSTREAMID     |                          |

## 6.1.1.39 \_\_spinClConfigurationEnums

```
enum __spinClConfigurationEnums
```

< This Camera Link specific feature describes the configuration used by the camera. It helps especially when a camera is capable of operation in a non-standard configuration, and when the features PixelSize, SensorDigitization, Taps, and DeviceTapGeometry do not provide enough information for interpretation of the image data provided by the camera.

## Enumerator

|                           |                                                                                                                                                                                                                                                                                                                               |
|---------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CIConfiguration_Base      | Standard base configuration described by the Camera Link standard.                                                                                                                                                                                                                                                            |
| CIConfiguration_Medium    | Standard medium configuration described by the Camera Link standard.                                                                                                                                                                                                                                                          |
| CIConfiguration_Full      | Standard full configuration described by the Camera Link standard.                                                                                                                                                                                                                                                            |
| CIConfiguration_DualBase  | The camera streams the data from multiple taps (that do not fit in the standard base configuration) through two Camera Link base ports. It is responsibility of the application or frame grabber to reconstruct the full image. Only one of the ports (fixed) serves as the "master" for serial communication and triggering. |
| CIConfiguration_EightyBit | Standard 80-bit configuration with 10 taps of 8 bits or 8 taps of 10 bits, as described by the Camera Link standard.                                                                                                                                                                                                          |
| NUM_CLCONFIGURATION       |                                                                                                                                                                                                                                                                                                                               |

## 6.1.1.40 \_spinClTimeSlotsCountEnums

```
enum _spinClTimeSlotsCountEnums
```

< This Camera Link specific feature describes the time multiplexing of the camera link connection to transfer more than the configuration allows, in one single clock.

## Enumerator

|                        |       |
|------------------------|-------|
| CITimeSlotsCount_One   | One   |
| CITimeSlotsCount_Two   | Two   |
| CITimeSlotsCount_Three | Three |
| NUM_CLTIMESLOTSCOUNT   |       |

## 6.1.1.41 \_spinColorTransformationSelectorEnums

```
enum _spinColorTransformationSelectorEnums
```

< Selects which Color Transformation module is controlled by the various Color Transformation features

## Enumerator

|                                      |  |
|--------------------------------------|--|
| ColorTransformationSelector_RGBtoRGB |  |
| ColorTransformationSelector_RGBtoYUV |  |
| NUM_COLORTRANSFORMATIONSELECTOR      |  |

## 6.1.1.42 \_spinColorTransformationValueSelectorEnums

```
enum _spinColorTransformationValueSelectorEnums
```

< Selects the Gain factor or Offset of the Transformation matrix to access in the selected Color Transformation module

#### Enumerator

|                                          |  |
|------------------------------------------|--|
| ColorTransformationValueSelector_Gain00  |  |
| ColorTransformationValueSelector_Gain01  |  |
| ColorTransformationValueSelector_Gain02  |  |
| ColorTransformationValueSelector_Gain10  |  |
| ColorTransformationValueSelector_Gain11  |  |
| ColorTransformationValueSelector_Gain12  |  |
| ColorTransformationValueSelector_Gain20  |  |
| ColorTransformationValueSelector_Gain21  |  |
| ColorTransformationValueSelector_Gain22  |  |
| ColorTransformationValueSelector_Offset0 |  |
| ColorTransformationValueSelector_Offset1 |  |
| ColorTransformationValueSelector_Offset2 |  |
| NUM_COLORTRANSFORMATIONVALUESELECTOR     |  |

#### 6.1.1.43 \_spinCounterEventActivationEnums

enum `_spinCounterEventActivationEnums`

< Selects the activation mode of the event to increment the Counter.

#### Enumerator

|                                    |  |
|------------------------------------|--|
| CounterEventActivation_LevelLow    |  |
| CounterEventActivation_LevelHigh   |  |
| CounterEventActivation_FallingEdge |  |
| CounterEventActivation_RisingEdge  |  |
| CounterEventActivation_AnyEdge     |  |
| NUM_COUNTEREVENTACTIVATION         |  |

#### 6.1.1.44 \_spinCounterEventSourceEnums

enum `_spinCounterEventSourceEnums`

< Selects the event that will increment the counter

#### Enumerator

|                            |         |
|----------------------------|---------|
| CounterEventSource_Off     | Off     |
| CounterEventSource_MHzTick | MHzTick |
| CounterEventSource_Line0   | Line0   |

## Enumerator

|                                     |                  |
|-------------------------------------|------------------|
| CounterEventSource_Line1            | Line1            |
| CounterEventSource_Line2            | Line2            |
| CounterEventSource_Line3            | Line3            |
| CounterEventSource_UserOutput0      | UserOutput0      |
| CounterEventSource_UserOutput1      | UserOutput1      |
| CounterEventSource_UserOutput2      | UserOutput2      |
| CounterEventSource_UserOutput3      | UserOutput3      |
| CounterEventSource_Counter0Start    | Counter0Start    |
| CounterEventSource_Counter1Start    | Counter1Start    |
| CounterEventSource_Counter0End      | Counter0End      |
| CounterEventSource_Counter1End      | Counter1End      |
| CounterEventSource_LogicBlock0      | LogicBlock0      |
| CounterEventSource_LogicBlock1      | LogicBlock1      |
| CounterEventSource_ExposureStart    | ExposureStart    |
| CounterEventSource_ExposureEnd      | ExposureEnd      |
| CounterEventSource_FrameTriggerWait | FrameTriggerWait |
| NUM_COUNTEREVENTSOURCE              |                  |

6.1.1.45 `_spinCounterResetActivationEnums`

```
enum _spinCounterResetActivationEnums
```

< Selects the Activation mode of the Counter Reset Source signal.

## Enumerator

|                                    |  |
|------------------------------------|--|
| CounterResetActivation_LevelLow    |  |
| CounterResetActivation_LevelHigh   |  |
| CounterResetActivation_FallingEdge |  |
| CounterResetActivation_RisingEdge  |  |
| CounterResetActivation_AnyEdge     |  |
| NUM_COUNTERRESETACTIVATION         |  |

6.1.1.46 `_spinCounterResetSourceEnums`

```
enum _spinCounterResetSourceEnums
```

< Selects the signal that will be the source to reset the Counter.

## Enumerator

|                        |     |
|------------------------|-----|
| CounterResetSource_Off | Off |
|------------------------|-----|

## Enumerator

|                                     |                  |
|-------------------------------------|------------------|
| CounterResetSource_Line0            | Line0            |
| CounterResetSource_Line1            | Line1            |
| CounterResetSource_Line2            | Line2            |
| CounterResetSource_Line3            | Line3            |
| CounterResetSource_UserOutput0      | UserOutput0      |
| CounterResetSource_UserOutput1      | UserOutput1      |
| CounterResetSource_UserOutput2      | UserOutput2      |
| CounterResetSource_UserOutput3      | UserOutput3      |
| CounterResetSource_Counter0Start    | Counter0Start    |
| CounterResetSource_Counter1Start    | Counter1Start    |
| CounterResetSource_Counter0End      | Counter0End      |
| CounterResetSource_Counter1End      | Counter1End      |
| CounterResetSource_LogicBlock0      | LogicBlock0      |
| CounterResetSource_LogicBlock1      | LogicBlock1      |
| CounterResetSource_ExposureStart    | ExposureStart    |
| CounterResetSource_ExposureEnd      | ExposureEnd      |
| CounterResetSource_FrameTriggerWait | FrameTriggerWait |
| NUM_COUNTERRESETSOURCE              |                  |

## 6.1.1.47 \_spinCounterSelectorEnums

```
enum _spinCounterSelectorEnums
```

< Selects which counter to configure

## Enumerator

|                          |  |
|--------------------------|--|
| CounterSelector_Counter0 |  |
| CounterSelector_Counter1 |  |
| NUM_COUNTERSELECTOR      |  |

## 6.1.1.48 \_spinCounterStatusEnums

```
enum _spinCounterStatusEnums
```

< Returns the current status of the Counter.

## Enumerator

|                                  |                                                     |
|----------------------------------|-----------------------------------------------------|
| CounterStatus_CounterIdle        | The counter is idle.                                |
| CounterStatus_CounterTriggerWait | The counter is waiting for a start trigger.         |
| CounterStatus_CounterActive      | The counter is counting for the specified duration. |
| CounterStatus_CounterCompleted   | The counter reached the CounterDuration count.      |
| CounterStatus_CounterOverflow    | The counter reached its maximum possible count.     |
| NUM_COUNTERSTATUS                |                                                     |

### 6.1.1.49 `_spinCounterTriggerActivationEnums`

enum `_spinCounterTriggerActivationEnums`

< Selects the activation mode of the trigger to start the Counter.

Enumerator

|                                                   |  |
|---------------------------------------------------|--|
| <code>CounterTriggerActivation_LevelLow</code>    |  |
| <code>CounterTriggerActivation_LevelHigh</code>   |  |
| <code>CounterTriggerActivation_FallingEdge</code> |  |
| <code>CounterTriggerActivation_RisingEdge</code>  |  |
| <code>CounterTriggerActivation_AnyEdge</code>     |  |
| <code>NUM_COUNTERTRIGGERACTIVATION</code>         |  |

### 6.1.1.50 `_spinCounterTriggerSourceEnums`

enum `_spinCounterTriggerSourceEnums`

< Selects the source of the trigger to start the counter

Enumerator

|                                                    |                  |
|----------------------------------------------------|------------------|
| <code>CounterTriggerSource_Off</code>              | Off              |
| <code>CounterTriggerSource_Line0</code>            | Line0            |
| <code>CounterTriggerSource_Line1</code>            | Line1            |
| <code>CounterTriggerSource_Line2</code>            | Line2            |
| <code>CounterTriggerSource_Line3</code>            | Line3            |
| <code>CounterTriggerSource_UserOutput0</code>      | UserOutput0      |
| <code>CounterTriggerSource_UserOutput1</code>      | UserOutput1      |
| <code>CounterTriggerSource_UserOutput2</code>      | UserOutput2      |
| <code>CounterTriggerSource_UserOutput3</code>      | UserOutput3      |
| <code>CounterTriggerSource_Counter0Start</code>    | Counter0Start    |
| <code>CounterTriggerSource_Counter1Start</code>    | Counter1Start    |
| <code>CounterTriggerSource_Counter0End</code>      | Counter0End      |
| <code>CounterTriggerSource_Counter1End</code>      | Counter1End      |
| <code>CounterTriggerSource_LogicBlock0</code>      | LogicBlock0      |
| <code>CounterTriggerSource_LogicBlock1</code>      | LogicBlock1      |
| <code>CounterTriggerSource_ExposureStart</code>    | ExposureStart    |
| <code>CounterTriggerSource_ExposureEnd</code>      | ExposureEnd      |
| <code>CounterTriggerSource_FrameTriggerWait</code> | FrameTriggerWait |
| <code>NUM_COUNTERTRIGGERSOURCE</code>              |                  |

**6.1.1.51 \_spinCxpConnectionTestModeEnums**

```
enum _spinCxpConnectionTestModeEnums
```

< Enables the test mode for an individual physical connection of the Device.

**Enumerator**

|                              |        |
|------------------------------|--------|
| CxpConnectionTestMode_Off    | Off    |
| CxpConnectionTestMode_Mode1  | Mode 1 |
| NUM_CXP_CONNECTION_TEST_MODE |        |

**6.1.1.52 \_spinCxpLinkConfigurationEnums**

```
enum _spinCxpLinkConfigurationEnums
```

< This feature allows specifying the Link configuration for the communication between the Receiver and Transmitter Device. In most cases this feature does not need to be written because automatic discovery will set configuration correctly to the value returned by CxpLinkConfigurationPreferred. Note that the currently active configuration of the Link can be read using CxpLinkConfigurationStatus.

**Enumerator**

|                              |                                                                        |
|------------------------------|------------------------------------------------------------------------|
| CxpLinkConfiguration_Auto    | Sets Automatic discovery for the Link Configuration.                   |
| CxpLinkConfiguration_CXP1_X1 | Force the Link to 1 Connection operating at CXP-1 speed (1.25 Gbps).   |
| CxpLinkConfiguration_CXP2_X1 | Force the Link to 1 Connection operating at CXP-2 speed (2.50 Gbps).   |
| CxpLinkConfiguration_CXP3_X1 | Force the Link to 1 Connection operating at CXP-3 speed (3.125 Gbps).  |
| CxpLinkConfiguration_CXP5_X1 | Force the Link to 1 Connection operating at CXP-5 speed (5.00 Gbps).   |
| CxpLinkConfiguration_CXP6_X1 | Force the Link to 1 Connection operating at CXP-6 speed (6.25 Gbps).   |
| CxpLinkConfiguration_CXP1_X2 | Force the Link to 2 Connections operating at CXP-1 speed (1.25 Gbps).  |
| CxpLinkConfiguration_CXP2_X2 | Force the Link to 2 Connections operating at CXP-2 speed (2.50 Gbps).  |
| CxpLinkConfiguration_CXP3_X2 | Force the Link to 2 Connections operating at CXP-3 speed (3.125 Gbps). |
| CxpLinkConfiguration_CXP5_X2 | Force the Link to 2 Connections operating at CXP-5 speed (5.00 Gbps).  |
| CxpLinkConfiguration_CXP6_X2 | Force the Link to 3 Connections operating at CXP-6 speed (6.25 Gbps).  |
| CxpLinkConfiguration_CXP1_X3 | Force the Link to 3 Connections operating at CXP-1 speed (1.25 Gbps).  |
| CxpLinkConfiguration_CXP2_X3 | Force the Link to 3 Connections operating at CXP-2 speed (2.50 Gbps).  |
| CxpLinkConfiguration_CXP3_X3 | Force the Link to 3 Connections operating at CXP-3 speed (3.125 Gbps). |
| CxpLinkConfiguration_CXP5_X3 | Force the Link to 3 Connections operating at CXP-5 speed (5.00 Gbps).  |
| CxpLinkConfiguration_CXP6_X3 | Force the Link to 3 Connections operating at CXP-6 speed (6.25 Gbps).  |
| CxpLinkConfiguration_CXP1_X4 | Force the Link to 4 Connections operating at CXP-1 speed (1.25 Gbps).  |
| CxpLinkConfiguration_CXP2_X4 | Force the Link to 4 Connections operating at CXP-2 speed (2.50 Gbps).  |
| CxpLinkConfiguration_CXP3_X4 | Force the Link to 4 Connections operating at CXP-3 speed (3.125 Gbps). |
| CxpLinkConfiguration_CXP5_X4 | Force the Link to 4 Connections operating at CXP-5 speed (5.00 Gbps).  |
| CxpLinkConfiguration_CXP6_X4 | Force the Link to 4 Connections operating at CXP-6 speed (6.25 Gbps).  |

## Enumerator

|                              |                                                                        |
|------------------------------|------------------------------------------------------------------------|
| CxpLinkConfiguration_CXP1_X5 | Force the Link to 5 Connections operating at CXP-1 speed (1.25 Gbps).  |
| CxpLinkConfiguration_CXP2_X5 | Force the Link to 5 Connections operating at CXP-2 speed (2.50 Gbps).  |
| CxpLinkConfiguration_CXP3_X5 | Force the Link to 5 Connections operating at CXP-3 speed (3.125 Gbps). |
| CxpLinkConfiguration_CXP5_X5 | Force the Link to 5 Connections operating at CXP-5 speed (5.00 Gbps).  |
| CxpLinkConfiguration_CXP6_X5 | Force the Link to 5 Connections operating at CXP-6 speed (6.25 Gbps).  |
| CxpLinkConfiguration_CXP1_X6 | Force the Link to 6 Connections operating at CXP-1 speed (1.25 Gbps).  |
| CxpLinkConfiguration_CXP2_X6 | Force the Link to 6 Connections operating at CXP-2 speed (2.50 Gbps).  |
| CxpLinkConfiguration_CXP3_X6 | Force the Link to 6 Connections operating at CXP-3 speed (3.125 Gbps). |
| CxpLinkConfiguration_CXP5_X6 | Force the Link to 6 Connections operating at CXP-5 speed (5.00 Gbps).  |
| CxpLinkConfiguration_CXP6_X6 | Force the Link to 6 Connections operating at CXP-6 speed (6.25 Gbps).  |
| NUM_CXPLINKCONFIGURATION     |                                                                        |

## 6.1.1.53 \_spinCxpLinkConfigurationPreferredEnums

```
enum _spinCxpLinkConfigurationPreferredEnums
```

< Provides the Link configuration that allows the Transmitter Device to operate in its default mode.

## Enumerator

|                                       |                                                      |
|---------------------------------------|------------------------------------------------------|
| CxpLinkConfigurationPreferred_CXP1_X1 | 1 Connection operating at CXP-1 speed (1.25 Gbps).   |
| CxpLinkConfigurationPreferred_CXP2_X1 | 1 Connection operating at CXP-2 speed (2.50 Gbps).   |
| CxpLinkConfigurationPreferred_CXP3_X1 | 1 Connection operating at CXP-3 speed (3.125 Gbps).  |
| CxpLinkConfigurationPreferred_CXP5_X1 | 1 Connection operating at CXP-5 speed (5.00 Gbps).   |
| CxpLinkConfigurationPreferred_CXP6_X1 | 1 Connection operating at CXP-6 speed (6.25 Gbps).   |
| CxpLinkConfigurationPreferred_CXP1_X2 | 2 Connections operating at CXP-1 speed (1.25 Gbps).  |
| CxpLinkConfigurationPreferred_CXP2_X2 | 2 Connections operating at CXP-2 speed (2.50 Gbps).  |
| CxpLinkConfigurationPreferred_CXP3_X2 | 2 Connections operating at CXP-3 speed (3.125 Gbps). |
| CxpLinkConfigurationPreferred_CXP5_X2 | 2 Connections operating at CXP-4 speed (5.00 Gbps).  |
| CxpLinkConfigurationPreferred_CXP6_X2 | 3 Connections operating at CXP-5 speed (6.25 Gbps).  |
| CxpLinkConfigurationPreferred_CXP1_X3 | 3 Connections operating at CXP-1 speed (1.25 Gbps).  |
| CxpLinkConfigurationPreferred_CXP2_X3 | 3 Connections operating at CXP-2 speed (2.50 Gbps).  |
| CxpLinkConfigurationPreferred_CXP3_X3 | 3 Connections operating at CXP-3 speed (3.125 Gbps). |
| CxpLinkConfigurationPreferred_CXP5_X3 | 3 Connections operating at CXP-5 speed (5.00 Gbps).  |
| CxpLinkConfigurationPreferred_CXP6_X3 | 3 Connections operating at CXP-6 speed (6.25 Gbps).  |
| CxpLinkConfigurationPreferred_CXP1_X4 | 4 Connections operating at CXP-1 speed (1.25 Gbps).  |
| CxpLinkConfigurationPreferred_CXP2_X4 | 4 Connections operating at CXP-2 speed (2.50 Gbps).  |
| CxpLinkConfigurationPreferred_CXP3_X4 | 4 Connections operating at CXP-3 speed (3.125 Gbps). |
| CxpLinkConfigurationPreferred_CXP5_X4 | 4 Connections operating at CXP-5 speed (5.00 Gbps).  |
| CxpLinkConfigurationPreferred_CXP6_X4 | 4 Connections operating at CXP-6 speed (6.25 Gbps).  |
| CxpLinkConfigurationPreferred_CXP1_X5 | 5 Connections operating at CXP-1 speed (1.25 Gbps).  |
| CxpLinkConfigurationPreferred_CXP2_X5 | 5 Connections operating at CXP-2 speed (2.50 Gbps).  |
| CxpLinkConfigurationPreferred_CXP3_X5 | 5 Connections operating at CXP-3 speed (3.125 Gbps). |
| CxpLinkConfigurationPreferred_CXP5_X5 | 5 Connections operating at CXP-5 speed (5.00 Gbps).  |



## Enumerator

|                                       |                                                      |
|---------------------------------------|------------------------------------------------------|
| CxpLinkConfigurationPreferred_CXP6_X5 | 5 Connections operating at CXP-6 speed (6.25 Gbps).  |
| CxpLinkConfigurationPreferred_CXP1_X6 | 6 Connections operating at CXP-1 speed (1.25 Gbps).  |
| CxpLinkConfigurationPreferred_CXP2_X6 | 6 Connections operating at CXP-2 speed (2.50 Gbps).  |
| CxpLinkConfigurationPreferred_CXP3_X6 | 6 Connections operating at CXP-3 speed (3.125 Gbps). |
| CxpLinkConfigurationPreferred_CXP5_X6 | 6 Connections operating at CXP-5 speed (5.00 Gbps).  |
| CxpLinkConfigurationPreferred_CXP6_X6 | 6 Connections operating at CXP-6 speed (6.25 Gbps).  |
| NUM_CXPLINKCONFIGURATIONPREFERRED     |                                                      |

## 6.1.1.54 \_spinCxpLinkConfigurationStatusEnums

```
enum _spinCxpLinkConfigurationStatusEnums
```

< This feature indicates the current and active Link configuration used by the Device.

## Enumerator

|                                    |                                                                                                                               |
|------------------------------------|-------------------------------------------------------------------------------------------------------------------------------|
| CxpLinkConfigurationStatus_None    | The Link configuration of the Device is unknown. Either the configuration operation has failed or there is nothing connected. |
| CxpLinkConfigurationStatus_Pending | The Device is in the process of configuring the Link. The Link cannot be used yet.                                            |
| CxpLinkConfigurationStatus_CXP1_X1 | 1 Connection operating at CXP-1 speed (1.25 Gbps).                                                                            |
| CxpLinkConfigurationStatus_CXP2_X1 | 1 Connection operating at CXP-2 speed (2.50 Gbps).                                                                            |
| CxpLinkConfigurationStatus_CXP3_X1 | 1 Connection operating at CXP-3 speed (3.125 Gbps).                                                                           |
| CxpLinkConfigurationStatus_CXP5_X1 | 1 Connection operating at CXP-5 speed (5.00 Gbps).                                                                            |
| CxpLinkConfigurationStatus_CXP6_X1 | 1 Connection operating at CXP-6 speed (6.25 Gbps).                                                                            |
| CxpLinkConfigurationStatus_CXP1_X2 | 2 Connections operating at CXP-1 speed (1.25 Gbps).                                                                           |
| CxpLinkConfigurationStatus_CXP2_X2 | 2 Connections operating at CXP-2 speed (2.50 Gbps).                                                                           |
| CxpLinkConfigurationStatus_CXP3_X2 | 2 Connections operating at CXP-3 speed (3.125 Gbps).                                                                          |
| CxpLinkConfigurationStatus_CXP5_X2 | 2 Connections operating at CXP-4 speed (5.00 Gbps).                                                                           |
| CxpLinkConfigurationStatus_CXP6_X2 | 3 Connections operating at CXP-5 speed (6.25 Gbps).                                                                           |
| CxpLinkConfigurationStatus_CXP1_X3 | 3 Connections operating at CXP-1 speed (1.25 Gbps).                                                                           |
| CxpLinkConfigurationStatus_CXP2_X3 | 3 Connections operating at CXP-2 speed (2.50 Gbps).                                                                           |
| CxpLinkConfigurationStatus_CXP3_X3 | 3 Connections operating at CXP-3 speed (3.125 Gbps).                                                                          |
| CxpLinkConfigurationStatus_CXP5_X3 | 3 Connections operating at CXP-5 speed (5.00 Gbps).                                                                           |
| CxpLinkConfigurationStatus_CXP6_X3 | 3 Connections operating at CXP-6 speed (6.25 Gbps).                                                                           |
| CxpLinkConfigurationStatus_CXP1_X4 | 4 Connections operating at CXP-1 speed (1.25 Gbps).                                                                           |
| CxpLinkConfigurationStatus_CXP2_X4 | 4 Connections operating at CXP-2 speed (2.50 Gbps).                                                                           |
| CxpLinkConfigurationStatus_CXP3_X4 | 4 Connections operating at CXP-3 speed (3.125 Gbps).                                                                          |
| CxpLinkConfigurationStatus_CXP5_X4 | 4 Connections operating at CXP-5 speed (5.00 Gbps).                                                                           |
| CxpLinkConfigurationStatus_CXP6_X4 | 4 Connections operating at CXP-6 speed (6.25 Gbps).                                                                           |
| CxpLinkConfigurationStatus_CXP1_X5 | 5 Connections operating at CXP-1 speed (1.25 Gbps).                                                                           |
| CxpLinkConfigurationStatus_CXP2_X5 | 5 Connections operating at CXP-2 speed (2.50 Gbps).                                                                           |
| CxpLinkConfigurationStatus_CXP3_X5 | 5 Connections operating at CXP-3 speed (3.125 Gbps).                                                                          |
| CxpLinkConfigurationStatus_CXP5_X5 | 5 Connections operating at CXP-5 speed (5.00 Gbps).                                                                           |

## Enumerator

|                                    |                                                      |
|------------------------------------|------------------------------------------------------|
| CxpLinkConfigurationStatus_CXP6_X5 | 5 Connections operating at CXP-6 speed (6.25 Gbps).  |
| CxpLinkConfigurationStatus_CXP1_X6 | 6 Connections operating at CXP-1 speed (1.25 Gbps).  |
| CxpLinkConfigurationStatus_CXP2_X6 | 6 Connections operating at CXP-2 speed (2.50 Gbps).  |
| CxpLinkConfigurationStatus_CXP3_X6 | 6 Connections operating at CXP-3 speed (3.125 Gbps). |
| CxpLinkConfigurationStatus_CXP5_X6 | 6 Connections operating at CXP-5 speed (5.00 Gbps).  |
| CxpLinkConfigurationStatus_CXP6_X6 | 6 Connections operating at CXP-6 speed (6.25 Gbps).  |
| NUM_CXPLINKCONFIGURATIONSTATUS     |                                                      |

6.1.1.55 `_spinCxpPoCxpStatusEnums`

```
enum _spinCxpPoCxpStatusEnums
```

< Returns the Power over CoaXPress (PoCXP) status of the Device.

## Enumerator

|                        |                                                         |
|------------------------|---------------------------------------------------------|
| CxpPoCxpStatus_Auto    | Normal automatic PoCXP operation.                       |
| CxpPoCxpStatus_Off     | PoCXP is forced off.                                    |
| CxpPoCxpStatus_Tripped | The Link has shut down because of an over-current trip. |
| NUM_CXPPOCXPSTATUS     |                                                         |

6.1.1.56 `_spinDecimationHorizontalModeEnums`

```
enum _spinDecimationHorizontalModeEnums
```

< The mode used to reduce the horizontal resolution when DecimationHorizontal is used. The current implementation only supports a single decimation mode: Discard. Average should be achieved via Binning.

## Enumerator

|                                  |                                                             |
|----------------------------------|-------------------------------------------------------------|
| DecimationHorizontalMode_Discard | The value of every Nth pixel is kept, others are discarded. |
| NUM_DECIMATIONHORIZONTALMODE     |                                                             |

6.1.1.57 `_spinDecimationSelectorEnums`

```
enum _spinDecimationSelectorEnums
```

< Selects which decimation layer is controlled by the DecimationHorizontal and DecimationVertical features.

## Enumerator

|                           |                                                                                                                                                                                                                                                   |
|---------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DecimationSelector_All    | The total amount of decimation to be performed on the captured image data.                                                                                                                                                                        |
| DecimationSelector_Sensor | The portion of decimation to be performed on the sensor directly. Currently this is the only decimation layer available and hence is identical to the "All" layer. All decimation modification should therefore be done via the "All" layer only. |
| NUM_DECIMATIONSELECTOR    |                                                                                                                                                                                                                                                   |

## 6.1.1.58 \_spinDecimationVerticalModeEnums

```
enum _spinDecimationVerticalModeEnums
```

< The mode used to reduce the vertical resolution when DecimationVertical is used. The current implementation only supports a single decimation mode: Discard. Average should be achieved via Binning.

## Enumerator

|                                |                                                             |
|--------------------------------|-------------------------------------------------------------|
| DecimationVerticalMode_Discard | The value of every Nth pixel is kept, others are discarded. |
| NUM_DECIMATIONVERTICALMODE     |                                                             |

## 6.1.1.59 \_spinDefectCorrectionModeEnums

```
enum _spinDefectCorrectionModeEnums
```

< Controls the method used for replacing defective pixels.

## Enumerator

|                                |                                                                                                                     |
|--------------------------------|---------------------------------------------------------------------------------------------------------------------|
| DefectCorrectionMode_Average   | Pixels are replaced with the average of their neighbours. This is the normal mode of operation.                     |
| DefectCorrectionMode_Highlight | Pixels are replaced with the maximum pixel value (i.e., 255 for 8-bit images). Can be used for debugging the table. |
| DefectCorrectionMode_Zero      | Pixels are replaced by the value zero. Can be used for testing the table.                                           |
| NUM_DEFECTCORRECTIONMODE       |                                                                                                                     |

## 6.1.1.60 \_spinDeinterlacingEnums

```
enum _spinDeinterlacingEnums
```

< Controls how the device performs de-interlacing.

**Enumerator**

|                               |                                                                                 |
|-------------------------------|---------------------------------------------------------------------------------|
| Deinterlacing_Off             | The device doesn't perform de-interlacing.                                      |
| Deinterlacing_LineDuplication | The device performs de-interlacing by outputting each line of each field twice. |
| Deinterlacing_Weave           | The device performs de-interlacing by interleaving the lines of all fields.     |
| NUM_DEINTERLACING             |                                                                                 |

**6.1.1.61 \_spinDeviceCharacterSetEnums**

```
enum _spinDeviceCharacterSetEnums
```

< Character set used by the strings of the device`s bootstrap registers.

**Enumerator**

|                          |  |
|--------------------------|--|
| DeviceCharacterSet_UTF8  |  |
| DeviceCharacterSet_ASCII |  |
| NUM_DEVICECHARACTERSET   |  |

**6.1.1.62 \_spinDeviceClockSelectorEnums**

```
enum _spinDeviceClockSelectorEnums
```

< Selects the clock frequency to access from the device.

**Enumerator**

|                                        |                                                     |
|----------------------------------------|-----------------------------------------------------|
| DeviceClockSelector_Sensor             | Clock frequency of the image sensor of the camera.  |
| DeviceClockSelector_SensorDigitization | Clock frequency of the camera A/D conversion stage. |
| DeviceClockSelector_CameraLink         | Frequency of the Camera Link clock.                 |
| NUM_DEVICECLOCKSELECTOR                |                                                     |

**6.1.1.63 \_spinDeviceConnectionStatusEnums**

```
enum _spinDeviceConnectionStatusEnums
```

< Indicates the status of the specified Connection.

**Enumerator**

|                                 |                           |
|---------------------------------|---------------------------|
| DeviceConnectionStatus_Active   | Connection is in use.     |
| DeviceConnectionStatus_Inactive | Connection is not in use. |
| NUM_DEVICECONNECTIONSTATUS      |                           |

#### 6.1.1.64 \_spinDeviceIndicatorModeEnums

enum `_spinDeviceIndicatorModeEnums`

< Controls the LED behaviour: Inactive (off), Active (current status), or Error Status (off unless an error occurs).

##### Enumerator

|                                 |  |
|---------------------------------|--|
| DeviceIndicatorMode_Inactive    |  |
| DeviceIndicatorMode_Active      |  |
| DeviceIndicatorMode_ErrorStatus |  |
| NUM_DEVICEINDICATORMODE         |  |

#### 6.1.1.65 \_spinDeviceLinkHeartbeatModeEnums

enum `_spinDeviceLinkHeartbeatModeEnums`

< Activate or deactivate the Link's heartbeat.

##### Enumerator

|                             |                              |
|-----------------------------|------------------------------|
| DeviceLinkHeartbeatMode_On  | Enables the Link heartbeat.  |
| DeviceLinkHeartbeatMode_Off | Disables the Link heartbeat. |
| NUM_DEVICELINKHEARTBEATMODE |                              |

#### 6.1.1.66 \_spinDeviceLinkThroughputLimitModeEnums

enum `_spinDeviceLinkThroughputLimitModeEnums`

< Controls if the DeviceLinkThroughputLimit is active. When disabled, lower level TL specific features are expected to control the throughput. When enabled, DeviceLinkThroughputLimit controls the overall throughput.

##### Enumerator

|                                   |                                                 |
|-----------------------------------|-------------------------------------------------|
| DeviceLinkThroughputLimitMode_On  | Enables the DeviceLinkThroughputLimit feature.  |
| DeviceLinkThroughputLimitMode_Off | Disables the DeviceLinkThroughputLimit feature. |
| NUM_DEVICELINKTHROUGHPUTLIMITMODE |                                                 |

#### 6.1.1.67 \_spinDevicePowerSupplySelectorEnums

enum `_spinDevicePowerSupplySelectorEnums`

< Selects the power supply source to control or read.

Enumerator

|                                    |  |
|------------------------------------|--|
| DevicePowerSupplySelector_External |  |
| NUM_DEVICEPOWERSUPPLYSELECTOR      |  |

#### 6.1.1.68 \_spinDeviceRegistersEndiannessEnums

enum `_spinDeviceRegistersEndiannessEnums`

< Endianness of the registers of the device.

Enumerator

|                                  |  |
|----------------------------------|--|
| DeviceRegistersEndianness_Little |  |
| DeviceRegistersEndianness_Big    |  |
| NUM_DEVICEREGISTERSENDIANNES     |  |

#### 6.1.1.69 \_spinDeviceScanTypeEnums

enum `_spinDeviceScanTypeEnums`

< Scan type of the sensor of the device.

Enumerator

|                         |  |
|-------------------------|--|
| DeviceScanType_Areascan |  |
| NUM_DEVICESCANTYPE      |  |

#### 6.1.1.70 \_spinDeviceSerialPortBaudRateEnums

enum `_spinDeviceSerialPortBaudRateEnums`

< This feature controls the baud rate used by the selected serial port.

## Enumerator

|                                     |                                   |
|-------------------------------------|-----------------------------------|
| DeviceSerialPortBaudRate_Baud9600   | Serial port speed of 9600 baud.   |
| DeviceSerialPortBaudRate_Baud19200  | Serial port speed of 19200 baud.  |
| DeviceSerialPortBaudRate_Baud38400  | Serial port speed of 38400 baud.  |
| DeviceSerialPortBaudRate_Baud57600  | Serial port speed of 57600 baud.  |
| DeviceSerialPortBaudRate_Baud115200 | Serial port speed of 115200 baud. |
| DeviceSerialPortBaudRate_Baud230400 | Serial port speed of 230400 baud. |
| DeviceSerialPortBaudRate_Baud460800 | Serial port speed of 460800 baud. |
| DeviceSerialPortBaudRate_Baud921600 | Serial port speed of 921600 baud. |
| NUM_DEVICESERIALPORTBAUDRATE        |                                   |

## 6.1.1.71 \_spinDeviceSerialPortSelectorEnums

```
enum _spinDeviceSerialPortSelectorEnums
```

< Selects which serial port of the device to control.

## Enumerator

|                                     |                                                       |
|-------------------------------------|-------------------------------------------------------|
| DeviceSerialPortSelector_CameraLink | Serial port associated to the Camera link connection. |
| NUM_DEVICESERIALPORTSELECTOR        |                                                       |

## 6.1.1.72 \_spinDeviceStreamChannelEndiannessEnums

```
enum _spinDeviceStreamChannelEndiannessEnums
```

< Endianness of multi-byte pixel data for this stream.

## Enumerator

|                                      |                                       |
|--------------------------------------|---------------------------------------|
| DeviceStreamChannelEndianness_Big    | Stream channel data is big Endian.    |
| DeviceStreamChannelEndianness_Little | Stream channel data is little Endian. |
| NUM_DEVICESTREAMCHANNELENDIANNES     |                                       |

## 6.1.1.73 \_spinDeviceStreamChannelTypeEnums

```
enum _spinDeviceStreamChannelTypeEnums
```

< Reports the type of the stream channel.

## Enumerator

|                                     |                                  |
|-------------------------------------|----------------------------------|
| DeviceStreamChannelType_Transmitter | Data stream transmitter channel. |
| DeviceStreamChannelType_Receiver    | Data stream receiver channel.    |
| NUM_DEVICESTREAMCHANNELTYPE         |                                  |

## 6.1.1.74 \_spinDeviceTapGeometryEnums

enum `_spinDeviceTapGeometryEnums`

< This device tap geometry feature describes the geometrical properties characterizing the taps of a camera as presented at the output of the device.

## Enumerator

|                                                    |                                  |
|----------------------------------------------------|----------------------------------|
| DeviceTapGeometry_Geometry_1X_1Y                   | Geometry_1X_1Y                   |
| DeviceTapGeometry_Geometry_1X2_1Y                  | Geometry_1X2_1Y                  |
| DeviceTapGeometry_Geometry_1X2_1Y2                 | Geometry_1X2_1Y2                 |
| DeviceTapGeometry_Geometry_2X_1Y                   | Geometry_2X_1Y                   |
| DeviceTapGeometry_Geometry_2X_1Y2Geometry_2XE_1Y   | Geometry_2X_1Y2Geometry_2XE_1Y   |
| DeviceTapGeometry_Geometry_2XE_1Y2                 | Geometry_2XE_1Y2                 |
| DeviceTapGeometry_Geometry_2XM_1Y                  | Geometry_2XM_1Y                  |
| DeviceTapGeometry_Geometry_2XM_1Y2                 | Geometry_2XM_1Y2                 |
| DeviceTapGeometry_Geometry_1X_1Y2                  | Geometry_1X_1Y2                  |
| DeviceTapGeometry_Geometry_1X_2YE                  | Geometry_1X_2YE                  |
| DeviceTapGeometry_Geometry_1X3_1Y                  | Geometry_1X3_1Y                  |
| DeviceTapGeometry_Geometry_3X_1Y                   | Geometry_3X_1Y                   |
| DeviceTapGeometry_Geometry_1X                      | Geometry_1X                      |
| DeviceTapGeometry_Geometry_1X2                     | Geometry_1X2                     |
| DeviceTapGeometry_Geometry_2X                      | Geometry_2X                      |
| DeviceTapGeometry_Geometry_2XE                     | Geometry_2XE                     |
| DeviceTapGeometry_Geometry_2XM                     | Geometry_2XM                     |
| DeviceTapGeometry_Geometry_1X3                     | Geometry_1X3                     |
| DeviceTapGeometry_Geometry_3X                      | Geometry_3X                      |
| DeviceTapGeometry_Geometry_1X4_1Y                  | Geometry_1X4_1Y                  |
| DeviceTapGeometry_Geometry_4X_1Y                   | Geometry_4X_1Y                   |
| DeviceTapGeometry_Geometry_2X2_1Y                  | Geometry_2X2_1Y                  |
| DeviceTapGeometry_Geometry_2X2E_1YGeometry_2X2M_1Y | Geometry_2X2E_1YGeometry_2X2M_1Y |
| DeviceTapGeometry_Geometry_1X2_2YE                 | Geometry_1X2_2YE                 |
| DeviceTapGeometry_Geometry_2X_2YE                  | Geometry_2X_2YE                  |
| DeviceTapGeometry_Geometry_2XE_2YE                 | Geometry_2XE_2YE                 |
| DeviceTapGeometry_Geometry_2XM_2YE                 | Geometry_2XM_2YE                 |
| DeviceTapGeometry_Geometry_1X4                     | Geometry_1X4                     |
| DeviceTapGeometry_Geometry_4X                      | Geometry_4X                      |
| DeviceTapGeometry_Geometry_2X2                     | Geometry_2X2                     |
| DeviceTapGeometry_Geometry_2X2E                    | Geometry_2X2E                    |



## Enumerator

|                                     |                   |
|-------------------------------------|-------------------|
| DeviceTapGeometry_Geometry_2X2M     | Geometry_2X2M     |
| DeviceTapGeometry_Geometry_1X8_1Y   | Geometry_1X8_1Y   |
| DeviceTapGeometry_Geometry_8X_1Y    | Geometry_8X_1Y    |
| DeviceTapGeometry_Geometry_4X2_1Y   | Geometry_4X2_1Y   |
| DeviceTapGeometry_Geometry_2X2E_2YE | Geometry_2X2E_2YE |
| DeviceTapGeometry_Geometry_1X8      | Geometry_1X8      |
| DeviceTapGeometry_Geometry_8X       | Geometry_8X       |
| DeviceTapGeometry_Geometry_4X2      | Geometry_4X2      |
| DeviceTapGeometry_Geometry_4X2E     | Geometry_4X2E     |
| DeviceTapGeometry_Geometry_4X2E_1Y  | Geometry_4X2E_1Y  |
| DeviceTapGeometry_Geometry_1X10_1Y  | Geometry_1X10_1Y  |
| DeviceTapGeometry_Geometry_10X_1Y   | Geometry_10X_1Y   |
| DeviceTapGeometry_Geometry_1X10     | Geometry_1X10     |
| DeviceTapGeometry_Geometry_10X      | Geometry_10X      |
| NUM_DEVICETAPGEOMETRY               |                   |

## 6.1.1.75 \_spinDeviceTemperatureSelectorEnums

```
enum _spinDeviceTemperatureSelectorEnums
```

< Selects the location within the device, where the temperature will be measured.

## Enumerator

|                                  |  |
|----------------------------------|--|
| DeviceTemperatureSelector_Sensor |  |
| NUM_DEVICETEMPERATURESELECTOR    |  |

## 6.1.1.76 \_spinDeviceTLTypeEnums

```
enum _spinDeviceTLTypeEnums
```

< Transport Layer type of the device.

## Enumerator

|                           |  |
|---------------------------|--|
| DeviceTLType_GigEVision   |  |
| DeviceTLType_CameraLink   |  |
| DeviceTLType_CameraLinkHS |  |
| DeviceTLType_CoaXPress    |  |
| DeviceTLType_USB3Vision   |  |
| DeviceTLType_Custom       |  |
| NUM_DEVICETLTYPE          |  |

### 6.1.1.77 \_spinDeviceTypeEnums

enum `_spinDeviceTypeEnums`

< Returns the device type.

#### Enumerator

|                        |                                                     |
|------------------------|-----------------------------------------------------|
| DeviceType_Transmitter | Data stream transmitter device.                     |
| DeviceType_Receiver    | Data stream receiver device.                        |
| DeviceType_Transceiver | Data stream receiver and transmitter device.        |
| DeviceType_Peripheral  | Controllable device (with no data stream handling). |
| NUM_DEVICETYPE         |                                                     |

### 6.1.1.78 \_spinEncoderModeEnums

enum `_spinEncoderModeEnums`

< Selects if the count of encoder uses FourPhase mode with jitter filtering or the HighResolution mode without jitter filtering.

#### Enumerator

|                            |                                                                                                                         |
|----------------------------|-------------------------------------------------------------------------------------------------------------------------|
| EncoderMode_FourPhase      | The counter increments or decrements 1 for every full quadrature cycle with jitter filtering.                           |
| EncoderMode_HighResolution | The counter increments or decrements every quadrature phase for high resolution counting, but without jitter filtering. |
| NUM_ENCODERMODE            |                                                                                                                         |

### 6.1.1.79 \_spinEncoderOutputModeEnums

enum `_spinEncoderOutputModeEnums`

< Selects the conditions for the Encoder interface to generate a valid Encoder output signal.

#### Enumerator

|                              |                                                                                                                                                                                                      |
|------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| EncoderOutputMode_Off        | No output pulse are generated.                                                                                                                                                                       |
| EncoderOutputMode_PositionUp | Output pulses are generated at all new positions in the positive direction. If the encoder reverses no output pulse are generated until it has again passed the position where the reversal started. |

## Enumerator

|                                 |                                                                                                                                                                                                      |
|---------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| EncoderOutputMode_PositionDown  | Output pulses are generated at all new positions in the negative direction. If the encoder reverses no output pulse are generated until it has again passed the position where the reversal started. |
| EncoderOutputMode_DirectionUp   | Output pulses are generated at all position increments in the positive direction while ignoring negative direction motion.                                                                           |
| EncoderOutputMode_DirectionDown | Output pulses are generated at all position increments in the negative direction while ignoring positive direction motion.                                                                           |
| EncoderOutputMode_Motion        | Output pulses are generated at all motion increments in both directions.                                                                                                                             |
| NUM_ENCODEROUTPUTMODE           |                                                                                                                                                                                                      |

## 6.1.1.80 \_spinEncoderResetActivationEnums

```
enum _spinEncoderResetActivationEnums
```

< Selects the Activation mode of the Encoder Reset Source signal.

## Enumerator

|                                    |                                                                          |
|------------------------------------|--------------------------------------------------------------------------|
| EncoderResetActivation_RisingEdge  | Resets the Encoder on the Rising Edge of the signal.                     |
| EncoderResetActivation_FallingEdge | Resets the Encoder on the Falling Edge of the signal.                    |
| EncoderResetActivation_AnyEdge     | Resets the Encoder on the Falling or rising Edge of the selected signal. |
| EncoderResetActivation_LevelHigh   | Resets the Encoder as long as the selected signal level is High.         |
| EncoderResetActivation_LevelLow    | Resets the Encoder as long as the selected signal level is Low.          |
| NUM_ENCODERRESETACTIVATION         |                                                                          |

## 6.1.1.81 \_spinEncoderResetSourceEnums

```
enum _spinEncoderResetSourceEnums
```

< Selects the signals that will be the source to reset the Encoder.

## Enumerator

|                                       |                                                       |
|---------------------------------------|-------------------------------------------------------|
| EncoderResetSource_Off                | Disable the Encoder Reset trigger.                    |
| EncoderResetSource_AcquisitionTrigger | Resets with the reception of the Acquisition Trigger. |
| EncoderResetSource_AcquisitionStart   | Resets with the reception of the Acquisition Start.   |
| EncoderResetSource_AcquisitionEnd     | Resets with the reception of the Acquisition End.     |
| EncoderResetSource_FrameTrigger       | Resets with the reception of the Frame Start Trigger. |
| EncoderResetSource_FrameStart         | Resets with the reception of the Frame Start.         |
| EncoderResetSource_FrameEnd           | Resets with the reception of the Frame End.           |
| EncoderResetSource_ExposureStart      | Resets with the reception of the Exposure Start.      |

## Enumerator

|                                    |                                                                                               |
|------------------------------------|-----------------------------------------------------------------------------------------------|
| EncoderResetSource_ExposureEnd     | Resets with the reception of the Exposure End.                                                |
| EncoderResetSource_Line0           | Resets by the chosen I/O Line.                                                                |
| EncoderResetSource_Line1           | Resets by the chosen I/O Line.                                                                |
| EncoderResetSource_Line2           | Resets by the chosen I/O Line.                                                                |
| EncoderResetSource_Counter0Start   | Resets with the reception of the Counter Start.                                               |
| EncoderResetSource_Counter1Start   | Resets with the reception of the Counter Start.                                               |
| EncoderResetSource_Counter2Start   | Resets with the reception of the Counter Start.                                               |
| EncoderResetSource_Counter0End     | Resets with the reception of the Counter End.                                                 |
| EncoderResetSource_Counter1End     | Resets with the reception of the Counter End.                                                 |
| EncoderResetSource_Counter2End     | Resets with the reception of the Counter End.                                                 |
| EncoderResetSource_Timer0Start     | Resets with the reception of the Timer Start.                                                 |
| EncoderResetSource_Timer1Start     | Resets with the reception of the Timer Start.                                                 |
| EncoderResetSource_Timer2Start     | Resets with the reception of the Timer Start.                                                 |
| EncoderResetSource_Timer0End       | Resets with the reception of the Timer End.                                                   |
| EncoderResetSource_Timer1End       | Resets with the reception of the Timer End.                                                   |
| EncoderResetSource_Timer2End       | Resets with the reception of the Timer End.                                                   |
| EncoderResetSource_UserOutput0     | Resets by the chosen User Output bit.                                                         |
| EncoderResetSource_UserOutput1     | Resets by the chosen User Output bit.                                                         |
| EncoderResetSource_UserOutput2     | Resets by the chosen User Output bit.                                                         |
| EncoderResetSource_SoftwareSignal0 | Resets on the reception of the Software Signal.                                               |
| EncoderResetSource_SoftwareSignal1 | Resets on the reception of the Software Signal.                                               |
| EncoderResetSource_SoftwareSignal2 | Resets on the reception of the Software Signal.                                               |
| EncoderResetSource_Action0         | Resets on assertions of the chosen action signal (Broadcasted signal on the transport layer). |
| EncoderResetSource_Action1         | Resets on assertions of the chosen action signal (Broadcasted signal on the transport layer). |
| EncoderResetSource_Action2         | Resets on assertions of the chosen action signal (Broadcasted signal on the transport layer). |
| EncoderResetSource_LinkTrigger0    | Resets on the reception of the chosen Link Trigger (received from the transport layer).       |
| EncoderResetSource_LinkTrigger1    | Resets on the reception of the chosen Link Trigger (received from the transport layer).       |
| EncoderResetSource_LinkTrigger2    | Resets on the reception of the chosen Link Trigger (received from the transport layer).       |
| NUM_ENCODERRESETSOURCE             |                                                                                               |

## 6.1.1.82 \_spinEncoderSelectorEnums

```
enum _spinEncoderSelectorEnums
```

< Selects which Encoder to configure.

## Enumerator

|                          |                    |
|--------------------------|--------------------|
| EncoderSelector_Encoder0 | Selects Encoder 0. |
| EncoderSelector_Encoder1 | Selects Encoder 1. |
| EncoderSelector_Encoder2 | Selects Encoder 2. |
| NUM_ENCODERSELECTOR      |                    |

**6.1.1.83 \_spinEncoderSourceAEnums**

```
enum _spinEncoderSourceAEnums
```

< Selects the signal which will be the source of the A input of the Encoder.

**Enumerator**

|                      |                                                          |
|----------------------|----------------------------------------------------------|
| EncoderSourceA_Off   | Counter is stopped.                                      |
| EncoderSourceA_Line0 | Encoder Forward input is taken from the chosen I/O Line. |
| EncoderSourceA_Line1 | Encoder Forward input is taken from the chosen I/O Line. |
| EncoderSourceA_Line2 | Encoder Forward input is taken from the chosen I/O Line. |
| NUM_ENCODERSOURCEA   |                                                          |

**6.1.1.84 \_spinEncoderSourceBEnums**

```
enum _spinEncoderSourceBEnums
```

< Selects the signal which will be the source of the B input of the Encoder.

**Enumerator**

|                      |                                                           |
|----------------------|-----------------------------------------------------------|
| EncoderSourceB_Off   | Counter is stopped.                                       |
| EncoderSourceB_Line0 | Encoder Reverse input is taken from the chosen I/O Line.. |
| EncoderSourceB_Line1 | Encoder Reverse input is taken from the chosen I/O Line.. |
| EncoderSourceB_Line2 | Encoder Reverse input is taken from the chosen I/O Line.. |
| NUM_ENCODERSOURCEB   |                                                           |

**6.1.1.85 \_spinEncoderStatusEnums**

```
enum _spinEncoderStatusEnums
```

< Returns the motion status of the encoder.

**Enumerator**

|                             |                                           |
|-----------------------------|-------------------------------------------|
| EncoderStatus_EncoderUp     | The encoder counter last incremented.     |
| EncoderStatus_EncoderDown   | The encoder counter last decremented.     |
| EncoderStatus_EncoderIdle   | The encoder is not active.                |
| EncoderStatus_EncoderStatic | No motion within the EncoderTimeout time. |
| NUM_ENCODERSTATUS           |                                           |

### 6.1.1.86 \_spinEventNotificationEnums

```
enum _spinEventNotificationEnums
```

< Enables/Disables the selected event.

Enumerator

|                       |  |
|-----------------------|--|
| EventNotification_On  |  |
| EventNotification_Off |  |
| NUM_EVENTNOTIFICATION |  |

### 6.1.1.87 \_spinEventSelectorEnums

```
enum _spinEventSelectorEnums
```

< Selects which Event to enable or disable.

Enumerator

|                                 |  |
|---------------------------------|--|
| EventSelector_Error             |  |
| EventSelector_ExposureEnd       |  |
| EventSelector_SerialPortReceive |  |
| NUM_EVENTSELECTOR               |  |

### 6.1.1.88 \_spinExposureActiveModeEnums

```
enum _spinExposureActiveModeEnums
```

< Control sensor active exposure mode.

Enumerator

|                              |  |
|------------------------------|--|
| ExposureActiveMode_Line1     |  |
| ExposureActiveMode_AnyPixels |  |
| ExposureActiveMode_AllPixels |  |
| NUM_EXPOSUREACTIVEMODE       |  |

**6.1.1.89 \_spinExposureAutoEnums**enum `_spinExposureAutoEnums`

&lt; Sets the automatic exposure mode

**Enumerator**

|                         |                                                                                                  |
|-------------------------|--------------------------------------------------------------------------------------------------|
| ExposureAuto_Off        | Exposure time is manually controlled using ExposureTime                                          |
| ExposureAuto_Once       | Exposure time is adapted once by the device. Once it has converged, it returns to the Off state. |
| ExposureAuto_Continuous | Exposure time is constantly adapted by the device to maximize the dynamic range.                 |
| NUM_EXPOSUREAUTO        |                                                                                                  |

**6.1.1.90 \_spinExposureModeEnums**enum `_spinExposureModeEnums`

&lt; Sets the operation mode of the Exposure.

**Enumerator**

|                           |                                                                                                                                                    |
|---------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
| ExposureMode_Timed        | Timed exposure. The exposure time is set using the ExposureTime or ExposureAuto features and the exposure starts with the FrameStart or LineStart. |
| ExposureMode_TriggerWidth | Uses the width of the current Frame trigger signal pulse to control the exposure time.                                                             |
| NUM_EXPOSUREMODE          |                                                                                                                                                    |

**6.1.1.91 \_spinExposureTimeModeEnums**enum `_spinExposureTimeModeEnums`

&lt; Sets the configuration mode of the ExposureTime feature.

**Enumerator**

|                             |                                                                                                                                                              |
|-----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ExposureTimeMode_Common     | The exposure time is common to all the color components. The common ExposureTime value to use can be set selecting it with ExposureTimeSelector[Common].     |
| ExposureTimeMode_Individual | The exposure time is individual for each color component. Each individual ExposureTime values to use can be set by selecting them with ExposureTimeSelector. |
| NUM_EXPOSURETIMEMODE        |                                                                                                                                                              |

### 6.1.1.92 \_spinExposureTimeSelectorEnums

```
enum _spinExposureTimeSelectorEnums
```

< Selects which exposure time is controlled by the ExposureTime feature. This allows for independent control over the exposure components.

#### Enumerator

|                                  |                                        |
|----------------------------------|----------------------------------------|
| ExposureTimeSelector_Common      | Selects the common ExposureTime.       |
| ExposureTimeSelector_Red         | Selects the red common ExposureTime.   |
| ExposureTimeSelector_Green       | Selects the green ExposureTime.        |
| ExposureTimeSelector_Blue        | Selects the blue ExposureTime.         |
| ExposureTimeSelector_Cyan        | Selects the cyan common ExposureTime.  |
| ExposureTimeSelector_Magenta     | Selects the magenta ExposureTime.      |
| ExposureTimeSelector_Yellow      | Selects the yellow ExposureTime.       |
| ExposureTimeSelector_Infrared    | Selects the infrared ExposureTime.     |
| ExposureTimeSelector_Ultraviolet | Selects the ultraviolet ExposureTime.  |
| ExposureTimeSelector_Stage1      | Selects the first stage ExposureTime.  |
| ExposureTimeSelector_Stage2      | Selects the second stage ExposureTime. |
| NUM_EXPOSURETIMESELECTOR         |                                        |

### 6.1.1.93 \_spinFileOpenModeEnums

```
enum _spinFileOpenModeEnums
```

< The mode of the file when it is opened. The file can be opened for reading, writing or both. This must be set before opening the file.

#### Enumerator

|                        |  |
|------------------------|--|
| FileOpenMode_Read      |  |
| FileOpenMode_Write     |  |
| FileOpenMode_ReadWrite |  |
| NUM_FILEOPENMODE       |  |

### 6.1.1.94 \_spinFileOperationSelectorEnums

```
enum _spinFileOperationSelectorEnums
```

< Sets operation to execute on the selected file when the execute command is given.



## Enumerator

|                              |  |
|------------------------------|--|
| FileOperationSelector_Open   |  |
| FileOperationSelector_Close  |  |
| FileOperationSelector_Read   |  |
| FileOperationSelector_Write  |  |
| FileOperationSelector_Delete |  |
| NUM_FILEOPERATIONSELECTOR    |  |

## 6.1.1.95 \_spinFileOperationStatusEnums

```
enum _spinFileOperationStatusEnums
```

< Represents the file operation execution status.

## Enumerator

|                              |                                                          |
|------------------------------|----------------------------------------------------------|
| FileOperationStatus_Success  | File Operation was successful.                           |
| FileOperationStatus_Failure  | File Operation failed.                                   |
| FileOperationStatus_Overflow | An overflow occurred while executing the File Operation. |
| NUM_FILEOPERATIONSTATUS      |                                                          |

## 6.1.1.96 \_spinFileSelectorEnums

```
enum _spinFileSelectorEnums
```

< Selects which file is being operated on. This must be set before performing any file operations.

## Enumerator

|                             |  |
|-----------------------------|--|
| FileSelector_UserSetDefault |  |
| FileSelector_UserSet0       |  |
| FileSelector_UserSet1       |  |
| FileSelector_UserFile1      |  |
| FileSelector_SerialPort0    |  |
| NUM_FILESELECTOR            |  |

## 6.1.1.97 \_spinGainAutoBalanceEnums

```
enum _spinGainAutoBalanceEnums
```

< Sets the mode for automatic gain balancing between the sensor color channels or taps. The gain coefficients of each channel or tap are adjusted so they are matched.

## Enumerator

|                            |                                                                                                                                    |
|----------------------------|------------------------------------------------------------------------------------------------------------------------------------|
| GainAutoBalance_Off        | Gain tap balancing is user controlled using Gain .                                                                                 |
| GainAutoBalance_Once       | Gain tap balancing is automatically adjusted once by the device. Once it has converged, it automatically returns to the Off state. |
| GainAutoBalance_Continuous | Gain tap balancing is constantly adjusted by the device.                                                                           |
| NUM_GAINAUTOBALANCE        |                                                                                                                                    |

## 6.1.1.98 \_spinGainAutoEnums

```
enum _spinGainAutoEnums
```

< Sets the automatic gain mode. Set to Off for manual control. Set to Once for a single automatic adjustment then return to Off. Set to Continuous for constant adjustment. In automatic modes, the camera adjusts the gain to maximize the dynamic range.

## Enumerator

|                     |                                                                                         |
|---------------------|-----------------------------------------------------------------------------------------|
| GainAuto_Off        | Gain is manually controlled                                                             |
| GainAuto_Once       | Gain is adapted once by the device. Once it has converged, it returns to the Off state. |
| GainAuto_Continuous | Gain is constantly adapted by the device to maximize the dynamic range.                 |
| NUM_GAINAUTO        |                                                                                         |

## 6.1.1.99 \_spinGainSelectorEnums

```
enum _spinGainSelectorEnums
```

< Selects which gain to control. The All selection is a total amplification across all channels (or taps).

## Enumerator

|                  |  |
|------------------|--|
| GainSelector_All |  |
| NUM_GAINSELECTOR |  |

## 6.1.1.100 \_spinGevCCPEnums

```
enum _spinGevCCPEnums
```

< Controls the device access privilege of an application.

**Enumerator**

|                        |  |
|------------------------|--|
| GevCCP_OpenAccess      |  |
| GevCCP_ExclusiveAccess |  |
| GevCCP_ControlAccess   |  |
| NUM_GEVCCP             |  |

**6.1.1.101 \_spinGevCurrentPhysicalLinkConfigurationEnums**

```
enum _spinGevCurrentPhysicalLinkConfigurationEnums
```

< Indicates the current physical link configuration of the device.

**Enumerator**

|                                                |             |
|------------------------------------------------|-------------|
| GevCurrentPhysicalLinkConfiguration_SingleLink | Single Link |
| GevCurrentPhysicalLinkConfiguration_MultiLink  | Multi Link  |
| GevCurrentPhysicalLinkConfiguration_StaticLAG  | Static LAG  |
| GevCurrentPhysicalLinkConfiguration_DynamicLAG | Dynamic LAG |
| NUM_GEVCURRENTPHYSICALLINKCONFIGURATION        |             |

**6.1.1.102 \_spinGevGVCPExtendedStatusCodesSelectorEnums**

```
enum _spinGevGVCPExtendedStatusCodesSelectorEnums
```

< Selects the GigE Vision version to control extended status codes for.

**Enumerator**

|                                               |             |
|-----------------------------------------------|-------------|
| GevGVCPExtendedStatusCodesSelector_Version1_1 | Version 1 1 |
| GevGVCPExtendedStatusCodesSelector_Version2_0 | Version 2 0 |
| NUM_GEVGVCPEXTENDEDSTATUSCODESSELECTOR        |             |

**6.1.1.103 \_spinGevGVSPExtendedIDModeEnums**

```
enum _spinGevGVSPExtendedIDModeEnums
```

< Enables the extended IDs mode.

## Enumerator

|                           |     |
|---------------------------|-----|
| GevGVSPExtendedIDMode_Off | Off |
| GevGVSPExtendedIDMode_On  | On  |
| NUM_GEVGVSPEXTENDEDIDMODE |     |

**6.1.1.104 \_spinGevIEEE1588ClockAccuracyEnums**

```
enum _spinGevIEEE1588ClockAccuracyEnums
```

< Indicates the expected accuracy of the device clock when it is the grandmaster, or in the event it becomes the grandmaster.

## Enumerator

|                                  |                  |
|----------------------------------|------------------|
| GevIEEE1588ClockAccuracy_Unknown | Unknown Accuracy |
| NUM_GEVIEEE1588CLOCKACCURACY     |                  |

**6.1.1.105 \_spinGevIEEE1588ModeEnums**

```
enum _spinGevIEEE1588ModeEnums
```

< Provides the mode of the IEEE 1588 clock.

## Enumerator

|                           |            |
|---------------------------|------------|
| GevIEEE1588Mode_Auto      | Automatic  |
| GevIEEE1588Mode_SlaveOnly | Slave Only |
| NUM_GEVIEEE1588MODE       |            |

**6.1.1.106 \_spinGevIEEE1588StatusEnums**

```
enum _spinGevIEEE1588StatusEnums
```

< Provides the status of the IEEE 1588 clock.

## Enumerator

|                                |              |
|--------------------------------|--------------|
| GevIEEE1588Status_Initializing | Initializing |
| GevIEEE1588Status_Faulty       | Faulty       |
| GevIEEE1588Status_Disabled     | Disabled     |

## Enumerator

|                                |              |
|--------------------------------|--------------|
| GevIEEE1588Status_Listening    | Listening    |
| GevIEEE1588Status_PreMaster    | Pre Master   |
| GevIEEE1588Status_Master       | Master       |
| GevIEEE1588Status_Passive      | Passive      |
| GevIEEE1588Status_Uncalibrated | Uncalibrated |
| GevIEEE1588Status_Slave        | Slave        |
| NUM_GEVIEEE1588STATUS          |              |

6.1.1.107 `_spinGevIPConfigurationStatusEnums`

enum `_spinGevIPConfigurationStatusEnums`

< Reports the current IP configuration status.

## Enumerator

|                                       |               |
|---------------------------------------|---------------|
| GevIPConfigurationStatus_None         | None          |
| GevIPConfigurationStatus_PersistentIP | Persistent IP |
| GevIPConfigurationStatus_DHCP         | DHCP          |
| GevIPConfigurationStatus_LLA          | LLA           |
| GevIPConfigurationStatus_ForceIP      | Force IP      |
| NUM_GEVIPCONFIGURATIONSTATUS          |               |

6.1.1.108 `_spinGevPhysicalLinkConfigurationEnums`

enum `_spinGevPhysicalLinkConfigurationEnums`

< Controls the principal physical link configuration to use on next restart/power-up of the device.

## Enumerator

|                                         |             |
|-----------------------------------------|-------------|
| GevPhysicalLinkConfiguration_SingleLink | Single Link |
| GevPhysicalLinkConfiguration_MultiLink  | Multi Link  |
| GevPhysicalLinkConfiguration_StaticLAG  | Static LAG  |
| GevPhysicalLinkConfiguration_DynamicLAG | Dynamic LAG |
| NUM_GEVPHYSICALLINKCONFIGURATION        |             |

**6.1.1.109 \_spinGevSupportedOptionSelectorEnums**

```
enum _spinGevSupportedOptionSelectorEnums
```

< Selects the GEV option to interrogate for existing support.

**Enumerator**

|                                                        |  |
|--------------------------------------------------------|--|
| GevSupportedOptionSelector_UserDefinedName             |  |
| GevSupportedOptionSelector_SerialNumber                |  |
| GevSupportedOptionSelector_HeartbeatDisable            |  |
| GevSupportedOptionSelector_LinkSpeed                   |  |
| GevSupportedOptionSelector_CCPApplicationSocket        |  |
| GevSupportedOptionSelector_ManifestTable               |  |
| GevSupportedOptionSelector_TestData                    |  |
| GevSupportedOptionSelector_DiscoveryAckDelay           |  |
| GevSupportedOptionSelector_DiscoveryAckDelayWritable   |  |
| GevSupportedOptionSelector_ExtendedStatusCodes         |  |
| GevSupportedOptionSelector_Action                      |  |
| GevSupportedOptionSelector_PendingAck                  |  |
| GevSupportedOptionSelector_EventData                   |  |
| GevSupportedOptionSelector_Event                       |  |
| GevSupportedOptionSelector_PacketResend                |  |
| GevSupportedOptionSelector_WriteMem                    |  |
| GevSupportedOptionSelector_CommandsConcatenation       |  |
| GevSupportedOptionSelector_IPConfigurationLLA          |  |
| GevSupportedOptionSelector_IPConfigurationDHCP         |  |
| GevSupportedOptionSelector_IPConfigurationPersistentIP |  |
| GevSupportedOptionSelector_StreamChannelSourceSocket   |  |
| GevSupportedOptionSelector_MessageChannelSourceSocket  |  |
| NUM_GEVSUPPORTEDOPTIONSELECTOR                         |  |

**6.1.1.110 \_spinImageComponentSelectorEnums**

```
enum _spinImageComponentSelectorEnums
```

< Selects a component to activate data streaming from.

**Enumerator**

|                                    |                                                                                                                                                                    |
|------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ImageComponentSelector_Intensity   | The acquisition of intensity of the reflected light is controlled.                                                                                                 |
| ImageComponentSelector_Color       | The acquisition of color of the reflected light is controlled                                                                                                      |
| ImageComponentSelector_Infrared    | The acquisition of non-visible infrared light is controlled.                                                                                                       |
| ImageComponentSelector_Ultraviolet | The acquisition of non-visible ultraviolet light is controlled.                                                                                                    |
| ImageComponentSelector_Range       | The acquisition of range (distance) data is controlled. The data produced may be only range (2.5D) or a point cloud 3D coordinates depending on the Scan3dControl. |

## Enumerator

|                                   |                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-----------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ImageComponentSelector_Disparity  | The acquisition of stereo camera disparity data is controlled. Disparity is a more specific range format approximately inversely proportional to distance. Disparity is typically given in pixel units.                                                                                                                                                                                                                         |
| ImageComponentSelector_Confidence | The acquisition of confidence map of the acquired image is controlled. Confidence data may be binary (0 - invalid) or an integer where 0 is invalid and increasing value is increased confidence in the data in the corresponding pixel. If floating point representation is used the confidence image is normalized to the range [0,1], for integer representation the maximum possible integer represents maximum confidence. |
| ImageComponentSelector_Scatter    | The acquisition of data measuring how much light is scattered around the reflected light. In processing this is used as an additional intensity image, often together with the standard intensity.                                                                                                                                                                                                                              |
| NUM_IMAGECOMPONENTSELECTOR        |                                                                                                                                                                                                                                                                                                                                                                                                                                 |

## 6.1.1.111 \_spinImageCompressionJPEGFormatOptionEnums

```
enum _spinImageCompressionJPEGFormatOptionEnums
```

< When JPEG is selected as the compression format, a device might optionally offer better control over JPEG-specific options through this feature.

## Enumerator

|                                                         |                                                                                                                                                                           |
|---------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ImageCompressionJPEGFormatOption_Lossless               | Selects lossless JPEG compression based on a predictive coding model.                                                                                                     |
| ImageCompressionJPEGFormatOption_Baseline↔<br>Standard  | Indicates this is a baseline sequential (single-scan) DCT-based JPEG.                                                                                                     |
| ImageCompressionJPEGFormatOption_Baseline↔<br>Optimized | Provides optimized color and slightly better compression than baseline standard by using custom Huffman tables optimized after statistical analysis of the image content. |
| ImageCompressionJPEGFormatOption_Progressive            | Indicates this is a progressive (multi-scan) DCT-based JPEG.                                                                                                              |
| NUM_IMAGECOMPRESSIONJPEGFORMATOPT↔<br>ION               |                                                                                                                                                                           |

## 6.1.1.112 \_spinImageCompressionModeEnums

```
enum _spinImageCompressionModeEnums
```

<



## Enumerator

|                               |  |
|-------------------------------|--|
| ImageCompressionMode_Off      |  |
| ImageCompressionMode_Lossless |  |
| NUM_IMAGECOMPRESSIONMODE      |  |

**6.1.1.113 \_spinImageCompressionRateOptionEnums**

```
enum _spinImageCompressionRateOptionEnums
```

< Two rate controlling options are offered: fixed bit rate or fixed quality. The exact implementation to achieve one or the other is vendor-specific.

## Enumerator

|                                       |                                                                                                                                                                   |
|---------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ImageCompressionRateOption_FixBitrate | Output stream follows a constant bit rate. Allows easy bandwidth management on the link.                                                                          |
| ImageCompressionRateOption_FixQuality | Output stream has a constant image quality. Can be used when image processing algorithms are sensitive to image degradation caused by excessive data compression. |
| NUM_IMAGECOMPRESSIONRATEOPTION        |                                                                                                                                                                   |

**6.1.1.114 \_spinLineFormatEnums**

```
enum _spinLineFormatEnums
```

< Displays the current electrical format of the selected physical input or output Line.

## Enumerator

|                        |  |
|------------------------|--|
| LineFormat_NoConnect   |  |
| LineFormat_TriState    |  |
| LineFormat_TTL         |  |
| LineFormat_LVDS        |  |
| LineFormat_RS422       |  |
| LineFormat_OptoCoupled |  |
| LineFormat_OpenDrain   |  |
| NUM_LINEFORMAT         |  |

**6.1.1.115 \_spinLineInputFilterSelectorEnums**

```
enum _spinLineInputFilterSelectorEnums
```

< Selects the kind of input filter to configure: Deglitch or Debounce.

#### Enumerator

|                                  |  |
|----------------------------------|--|
| LineInputFilterSelector_Deglitch |  |
| LineInputFilterSelector_Debounce |  |
| NUM_LINEINPUTFILTERSELECTOR      |  |

### 6.1.1.116 `_spinLineModeEnums`

enum `_spinLineModeEnums`

< Controls if the physical Line is used to Input or Output a signal.

#### Enumerator

|                 |  |
|-----------------|--|
| LineMode_Input  |  |
| LineMode_Output |  |
| NUM_LINEMODE    |  |

### 6.1.1.117 `_spinLineSelectorEnums`

enum `_spinLineSelectorEnums`

< Selects the physical line (or pin) of the external device connector to configure

#### Enumerator

|                    |  |
|--------------------|--|
| LineSelector_Line0 |  |
| LineSelector_Line1 |  |
| LineSelector_Line2 |  |
| LineSelector_Line3 |  |
| NUM_LINESELECTOR   |  |

### 6.1.1.118 `_spinLineSourceEnums`

enum `_spinLineSourceEnums`

< Selects which internal acquisition or I/O source signal to output on the selected line. LineMode must be Output.

## Enumerator

|                             |  |
|-----------------------------|--|
| LineSource_Off              |  |
| LineSource_Line0            |  |
| LineSource_Line1            |  |
| LineSource_Line2            |  |
| LineSource_Line3            |  |
| LineSource_UserOutput0      |  |
| LineSource_UserOutput1      |  |
| LineSource_UserOutput2      |  |
| LineSource_UserOutput3      |  |
| LineSource_Counter0Active   |  |
| LineSource_Counter1Active   |  |
| LineSource_LogicBlock0      |  |
| LineSource_LogicBlock1      |  |
| LineSource_ExposureActive   |  |
| LineSource_FrameTriggerWait |  |
| LineSource_SerialPort0      |  |
| LineSource_PPSSignal        |  |
| LineSource_AllPixel         |  |
| LineSource_AnyPixel         |  |
| NUM_LINESOURCE              |  |

## 6.1.1.119 \_spinLogicBlockLUTInputActivationEnums

```
enum _spinLogicBlockLUTInputActivationEnums
```

< Selects the activation mode of the Logic Input Source signal.

## Enumerator

|                                          |  |
|------------------------------------------|--|
| LogicBlockLUTInputActivation_LevelLow    |  |
| LogicBlockLUTInputActivation_LevelHigh   |  |
| LogicBlockLUTInputActivation_FallingEdge |  |
| LogicBlockLUTInputActivation_RisingEdge  |  |
| LogicBlockLUTInputActivation_AnyEdge     |  |
| NUM_LOGICBLOCKLUTINPUTACTIVATION         |  |

## 6.1.1.120 \_spinLogicBlockLUTInputSelectorEnums

```
enum _spinLogicBlockLUTInputSelectorEnums
```

< Controls which LogicBlockLUT Input Source & Activation to access.

## Enumerator

|                                   |  |
|-----------------------------------|--|
| LogicBlockLUTInputSelector_Input0 |  |
| LogicBlockLUTInputSelector_Input1 |  |
| LogicBlockLUTInputSelector_Input2 |  |
| LogicBlockLUTInputSelector_Input3 |  |
| NUM_LOGICBLOCKLUTINPUTSELECTOR    |  |

6.1.1.121 `_spinLogicBlockLUTInputSourceEnums`

```
enum _spinLogicBlockLUTInputSourceEnums
```

< Selects the source for the input into the Logic LUT.

## Enumerator

|                                            |                   |
|--------------------------------------------|-------------------|
| LogicBlockLUTInputSource_Zero              | Zero              |
| LogicBlockLUTInputSource_Line0             | Line0             |
| LogicBlockLUTInputSource_Line1             | Line1             |
| LogicBlockLUTInputSource_Line2             | Line2             |
| LogicBlockLUTInputSource_Line3             | Line3             |
| LogicBlockLUTInputSource_UserOutput0       | UserOutput0       |
| LogicBlockLUTInputSource_UserOutput1       | UserOutput1       |
| LogicBlockLUTInputSource_UserOutput2       | UserOutput2       |
| LogicBlockLUTInputSource_UserOutput3       | UserOutput3       |
| LogicBlockLUTInputSource_Counter0Start     | Counter0Start     |
| LogicBlockLUTInputSource_Counter1Start     | Counter1Start     |
| LogicBlockLUTInputSource_Counter0End       | Counter0End       |
| LogicBlockLUTInputSource_Counter1End       | Counter1End       |
| LogicBlockLUTInputSource_LogicBlock0       | LogicBlock0       |
| LogicBlockLUTInputSource_LogicBlock1       | LogicBlock1       |
| LogicBlockLUTInputSource_ExposureStart     | ExposureStart     |
| LogicBlockLUTInputSource_ExposureEnd       | ExposureEnd       |
| LogicBlockLUTInputSource_FrameTriggerWait  | FrameTriggerWait  |
| LogicBlockLUTInputSource_AcquisitionActive | AcquisitionActive |
| NUM_LOGICBLOCKLUTINPUTSOURCE               |                   |

6.1.1.122 `_spinLogicBlockLUTSelectorEnums`

```
enum _spinLogicBlockLUTSelectorEnums
```

< Selects which LogicBlock LUT to configure

## Enumerator

|                              |  |
|------------------------------|--|
| LogicBlockLUTSelector_Value  |  |
| LogicBlockLUTSelector_Enable |  |
| NUM_LOGICBLOCKLUTSELECTOR    |  |

**6.1.1.123 \_spinLogicBlockSelectorEnums**

```
enum _spinLogicBlockSelectorEnums
```

< Selects which LogicBlock to configure

## Enumerator

|                                |  |
|--------------------------------|--|
| LogicBlockSelector_LogicBlock0 |  |
| LogicBlockSelector_LogicBlock1 |  |
| NUM_LOGICBLOCKSELECTOR         |  |

**6.1.1.124 \_spinLUTSelectorEnums**

```
enum _spinLUTSelectorEnums
```

The enum definitions for camera nodes.

< Selects which LUT to control.

## Enumerator

|                  |                                                                                      |
|------------------|--------------------------------------------------------------------------------------|
| LUTSelector_LUT1 | This LUT is for re-mapping pixels of all formats (mono, Bayer, red, green and blue). |
| NUM_LUTSELECTOR  |                                                                                      |

**6.1.1.125 \_spinPixelColorFilterEnums**

```
enum _spinPixelColorFilterEnums
```

< Type of color filter that is applied to the image. Only applies to Bayer pixel formats. All others have no color filter.

## Enumerator

|                          |                         |
|--------------------------|-------------------------|
| PixelColorFilter_None    | No color filter.        |
| PixelColorFilter_BayerRG | Bayer Red Green filter. |

## Enumerator

|                          |                          |
|--------------------------|--------------------------|
| PixelColorFilter_BayerGB | Bayer Green Blue filter. |
| PixelColorFilter_BayerGR | Bayer Green Red filter.  |
| PixelColorFilter_BayerBG | Bayer Blue Green filter. |
| NUM_PIXELCOLORFILTER     |                          |

6.1.1.126 `_spinPixelFormatEnums`

```
enum _spinPixelFormatEnums
```

< Format of the pixel provided by the camera.

## Enumerator

|                             |  |
|-----------------------------|--|
| PixelFormat_Mono8           |  |
| PixelFormat_Mono16          |  |
| PixelFormat_RGB8Packed      |  |
| PixelFormat_BayerGR8        |  |
| PixelFormat_BayerRG8        |  |
| PixelFormat_BayerGB8        |  |
| PixelFormat_BayerBG8        |  |
| PixelFormat_BayerGR16       |  |
| PixelFormat_BayerRG16       |  |
| PixelFormat_BayerGB16       |  |
| PixelFormat_BayerBG16       |  |
| PixelFormat_Mono12Packed    |  |
| PixelFormat_BayerGR12Packed |  |
| PixelFormat_BayerRG12Packed |  |
| PixelFormat_BayerGB12Packed |  |
| PixelFormat_BayerBG12Packed |  |
| PixelFormat_YUV411Packed    |  |
| PixelFormat_YUV422Packed    |  |
| PixelFormat_YUV444Packed    |  |
| PixelFormat_Mono12p         |  |
| PixelFormat_BayerGR12p      |  |
| PixelFormat_BayerRG12p      |  |
| PixelFormat_BayerGB12p      |  |
| PixelFormat_BayerBG12p      |  |
| PixelFormat_YCbCr8          |  |
| PixelFormat_YCbCr422_8      |  |
| PixelFormat_YCbCr411_8      |  |
| PixelFormat_BGR8            |  |
| PixelFormat_BGRa8           |  |
| PixelFormat_Mono10Packed    |  |
| PixelFormat_BayerGR10Packed |  |
| PixelFormat_BayerRG10Packed |  |

## Enumerator

|                             |                                          |
|-----------------------------|------------------------------------------|
| PixelFormat_BayerGB10Packed |                                          |
| PixelFormat_BayerBG10Packed |                                          |
| PixelFormat_Mono10p         |                                          |
| PixelFormat_BayerGR10p      |                                          |
| PixelFormat_BayerRG10p      |                                          |
| PixelFormat_BayerGB10p      |                                          |
| PixelFormat_BayerBG10p      |                                          |
| PixelFormat_Mono1p          | Monochrome 1-bit packed                  |
| PixelFormat_Mono2p          | Monochrome 2-bit packed                  |
| PixelFormat_Mono4p          | Monochrome 4-bit packed                  |
| PixelFormat_Mono8s          | Monochrome 8-bit signed                  |
| PixelFormat_Mono10          | Monochrome 10-bit unpacked               |
| PixelFormat_Mono12          | Monochrome 12-bit unpacked               |
| PixelFormat_Mono14          | Monochrome 14-bit unpacked               |
| PixelFormat_Mono16s         | Monochrome 16-bit signed                 |
| PixelFormat_Mono32f         | Monochrome 32-bit float                  |
| PixelFormat_BayerBG10       | Bayer Blue-Green 10-bit unpacked         |
| PixelFormat_BayerBG12       | Bayer Blue-Green 12-bit unpacked         |
| PixelFormat_BayerGB10       | Bayer Green-Blue 10-bit unpacked         |
| PixelFormat_BayerGB12       | Bayer Green-Blue 12-bit unpacked         |
| PixelFormat_BayerGR10       | Bayer Green-Red 10-bit unpacked          |
| PixelFormat_BayerGR12       | Bayer Green-Red 12-bit unpacked          |
| PixelFormat_BayerRG10       | Bayer Red-Green 10-bit unpacked          |
| PixelFormat_BayerRG12       | Bayer Red-Green 12-bit unpacked          |
| PixelFormat_RGBa8           | Red-Green-Blue-alpha 8-bit               |
| PixelFormat_RGBa10          | Red-Green-Blue-alpha 10-bit unpacked     |
| PixelFormat_RGBa10p         | Red-Green-Blue-alpha 10-bit packed       |
| PixelFormat_RGBa12          | Red-Green-Blue-alpha 12-bit unpacked     |
| PixelFormat_RGBa12p         | Red-Green-Blue-alpha 12-bit packed       |
| PixelFormat_RGBa14          | Red-Green-Blue-alpha 14-bit unpacked     |
| PixelFormat_RGBa16          | Red-Green-Blue-alpha 16-bit              |
| PixelFormat_RGB8            | Red-Green-Blue 8-bit                     |
| PixelFormat_RGB8_Planar     | Red-Green-Blue 8-bit planar              |
| PixelFormat_RGB10           | Red-Green-Blue 10-bit unpacked           |
| PixelFormat_RGB10_Planar    | Red-Green-Blue 10-bit unpacked planar    |
| PixelFormat_RGB10p          | Red-Green-Blue 10-bit packed             |
| PixelFormat_RGB10p32        | Red-Green-Blue 10-bit packed into 32-bit |
| PixelFormat_RGB12           | Red-Green-Blue 12-bit unpacked           |
| PixelFormat_RGB12_Planar    | Red-Green-Blue 12-bit unpacked planar    |
| PixelFormat_RGB12p          | Red-Green-Blue 12-bit packed             |
| PixelFormat_RGB14           | Red-Green-Blue 14-bit unpacked           |
| PixelFormat_RGB16           | Red-Green-Blue 16-bit                    |
| PixelFormat_RGB16s          | Red-Green-Blue 16-bit signed             |
| PixelFormat_RGB32f          | Red-Green-Blue 32-bit float              |
| PixelFormat_RGB16_Planar    | Red-Green-Blue 16-bit planar             |

## Enumerator

|                                   |                                                  |
|-----------------------------------|--------------------------------------------------|
| PixelFormat_RGB565p               | Red-Green-Blue 5/6/5-bit packed                  |
| PixelFormat_BGRa10                | Blue-Green-Red-alpha 10-bit unpacked             |
| PixelFormat_BGRa10p               | Blue-Green-Red-alpha 10-bit packed               |
| PixelFormat_BGRa12                | Blue-Green-Red-alpha 12-bit unpacked             |
| PixelFormat_BGRa12p               | Blue-Green-Red-alpha 12-bit packed               |
| PixelFormat_BGRa14                | Blue-Green-Red-alpha 14-bit unpacked             |
| PixelFormat_BGRa16                | Blue-Green-Red-alpha 16-bit                      |
| PixelFormat_RGBa32f               | Red-Green-Blue-alpha 32-bit float                |
| PixelFormat_BGR10                 | Blue-Green-Red 10-bit unpacked                   |
| PixelFormat_BGR10p                | Blue-Green-Red 10-bit packed                     |
| PixelFormat_BGR12                 | Blue-Green-Red 12-bit unpacked                   |
| PixelFormat_BGR12p                | Blue-Green-Red 12-bit packed                     |
| PixelFormat_BGR14                 | Blue-Green-Red 14-bit unpacked                   |
| PixelFormat_BGR16                 | Blue-Green-Red 16-bit                            |
| PixelFormat_BGR565p               | Blue-Green-Red 5/6/5-bit packed                  |
| PixelFormat_R8                    | Red 8-bit                                        |
| PixelFormat_R10                   | Red 10-bit                                       |
| PixelFormat_R12                   | Red 12-bit                                       |
| PixelFormat_R16                   | Red 16-bit                                       |
| PixelFormat_G8                    | Green 8-bit                                      |
| PixelFormat_G10                   | Green 10-bit                                     |
| PixelFormat_G12                   | Green 12-bit                                     |
| PixelFormat_G16                   | Green 16-bit                                     |
| PixelFormat_B8                    | Blue 8-bit                                       |
| PixelFormat_B10                   | Blue 10-bit                                      |
| PixelFormat_B12                   | Blue 12-bit                                      |
| PixelFormat_B16                   | Blue 16-bit                                      |
| PixelFormat_Coord3D_ABC8          | 3D coordinate A-B-C 8-bit                        |
| PixelFormat_Coord3D_ABC8_Planar   | 3D coordinate A-B-C 8-bit planar                 |
| PixelFormat_Coord3D_ABC10p        | 3D coordinate A-B-C 10-bit packed                |
| PixelFormat_Coord3D_ABC10p_Planar | 3D coordinate A-B-C 10-bit packed planar         |
| PixelFormat_Coord3D_ABC12p        | 3D coordinate A-B-C 12-bit packed                |
| PixelFormat_Coord3D_ABC12p_Planar | 3D coordinate A-B-C 12-bit packed planar         |
| PixelFormat_Coord3D_ABC16         | 3D coordinate A-B-C 16-bit                       |
| PixelFormat_Coord3D_ABC16_Planar  | 3D coordinate A-B-C 16-bit planar                |
| PixelFormat_Coord3D_ABC32f        | 3D coordinate A-B-C 32-bit floating point        |
| PixelFormat_Coord3D_ABC32f_Planar | 3D coordinate A-B-C 32-bit floating point planar |
| PixelFormat_Coord3D_AC8           | 3D coordinate A-C 8-bit                          |
| PixelFormat_Coord3D_AC8_Planar    | 3D coordinate A-C 8-bit planar                   |
| PixelFormat_Coord3D_AC10p         | 3D coordinate A-C 10-bit packed                  |
| PixelFormat_Coord3D_AC10p_Planar  | 3D coordinate A-C 10-bit packed planar           |
| PixelFormat_Coord3D_AC12p         | 3D coordinate A-C 12-bit packed                  |
| PixelFormat_Coord3D_AC12p_Planar  | 3D coordinate A-C 12-bit packed planar           |
| PixelFormat_Coord3D_AC16          | 3D coordinate A-C 16-bit                         |
| PixelFormat_Coord3D_AC16_Planar   | 3D coordinate A-C 16-bit planar                  |
| PixelFormat_Coord3D_AC32f         | 3D coordinate A-C 32-bit floating point          |
| PixelFormat_Coord3D_AC32f_Planar  | 3D coordinate A-C 32-bit floating point planar   |



## Enumerator

|                            |                                                               |
|----------------------------|---------------------------------------------------------------|
| PixelFormat_Coord3D_A8     | 3D coordinate A 8-bit                                         |
| PixelFormat_Coord3D_A10p   | 3D coordinate A 10-bit packed                                 |
| PixelFormat_Coord3D_A12p   | 3D coordinate A 12-bit packed                                 |
| PixelFormat_Coord3D_A16    | 3D coordinate A 16-bit                                        |
| PixelFormat_Coord3D_A32f   | 3D coordinate A 32-bit floating point                         |
| PixelFormat_Coord3D_B8     | 3D coordinate B 8-bit                                         |
| PixelFormat_Coord3D_B10p   | 3D coordinate B 10-bit packed                                 |
| PixelFormat_Coord3D_B12p   | 3D coordinate B 12-bit packed                                 |
| PixelFormat_Coord3D_B16    | 3D coordinate B 16-bit                                        |
| PixelFormat_Coord3D_B32f   | 3D coordinate B 32-bit floating point                         |
| PixelFormat_Coord3D_C8     | 3D coordinate C 8-bit                                         |
| PixelFormat_Coord3D_C10p   | 3D coordinate C 10-bit packed                                 |
| PixelFormat_Coord3D_C12p   | 3D coordinate C 12-bit packed                                 |
| PixelFormat_Coord3D_C16    | 3D coordinate C 16-bit                                        |
| PixelFormat_Coord3D_C32f   | 3D coordinate C 32-bit floating point                         |
| PixelFormat_Confidence1    | Confidence 1-bit unpacked                                     |
| PixelFormat_Confidence1p   | Confidence 1-bit packed                                       |
| PixelFormat_Confidence8    | Confidence 8-bit                                              |
| PixelFormat_Confidence16   | Confidence 16-bit                                             |
| PixelFormat_Confidence32f  | Confidence 32-bit floating point                              |
| PixelFormat_BiColorBGRG8   | Bi-color Blue/Green - Red/Green 8-bit                         |
| PixelFormat_BiColorBGRG10  | Bi-color Blue/Green - Red/Green 10-bit unpacked               |
| PixelFormat_BiColorBGRG10p | Bi-color Blue/Green - Red/Green 10-bit packed                 |
| PixelFormat_BiColorBGRG12  | Bi-color Blue/Green - Red/Green 12-bit unpacked               |
| PixelFormat_BiColorBGRG12p | Bi-color Blue/Green - Red/Green 12-bit packed                 |
| PixelFormat_BiColorRGBG8   | Bi-color Red/Green - Blue/Green 8-bit                         |
| PixelFormat_BiColorRGBG10  | Bi-color Red/Green - Blue/Green 10-bit unpacked               |
| PixelFormat_BiColorRGBG10p | Bi-color Red/Green - Blue/Green 10-bit packed                 |
| PixelFormat_BiColorRGBG12  | Bi-color Red/Green - Blue/Green 12-bit unpacked               |
| PixelFormat_BiColorRGBG12p | Bi-color Red/Green - Blue/Green 12-bit packed                 |
| PixelFormat_SCF1WBWG8      | Sparse Color Filter #1 White-Blue-White-Green 8-bit           |
| PixelFormat_SCF1WBWG10     | Sparse Color Filter #1 White-Blue-White-Green 10-bit unpacked |
| PixelFormat_SCF1WBWG10p    | Sparse Color Filter #1 White-Blue-White-Green 10-bit packed   |
| PixelFormat_SCF1WBWG12     | Sparse Color Filter #1 White-Blue-White-Green 12-bit unpacked |
| PixelFormat_SCF1WBWG12p    | Sparse Color Filter #1 White-Blue-White-Green 12-bit packed   |
| PixelFormat_SCF1WBWG14     | Sparse Color Filter #1 White-Blue-White-Green 14-bit unpacked |
| PixelFormat_SCF1WBWG16     | Sparse Color Filter #1 White-Blue-White-Green 16-bit unpacked |
| PixelFormat_SCF1WGWB8      | Sparse Color Filter #1 White-Green-White-Blue 8-bit           |
| PixelFormat_SCF1WGWB10     | Sparse Color Filter #1 White-Green-White-Blue 10-bit unpacked |
| PixelFormat_SCF1WGWB10p    | Sparse Color Filter #1 White-Green-White-Blue 10-bit packed   |
| PixelFormat_SCF1WGWB12     | Sparse Color Filter #1 White-Green-White-Blue 12-bit unpacked |
| PixelFormat_SCF1WGWB12p    | Sparse Color Filter #1 White-Green-White-Blue 12-bit packed   |
| PixelFormat_SCF1WGWB14     | Sparse Color Filter #1 White-Green-White-Blue 14-bit unpacked |
| PixelFormat_SCF1WGWB16     | Sparse Color Filter #1 White-Green-White-Blue 16-bit          |
| PixelFormat_SCF1WGWR8      | Sparse Color Filter #1 White-Green-White-Red 8-bit            |
| PixelFormat_SCF1WGWR10     | Sparse Color Filter #1 White-Green-White-Red 10-bit unpacked  |
| PixelFormat_SCF1WGWR10p    | Sparse Color Filter #1 White-Green-White-Red 10-bit packed    |

## Enumerator

|                                     |                                                              |
|-------------------------------------|--------------------------------------------------------------|
| PixelFormat_SCF1WGWR12              | Sparse Color Filter #1 White-Green-White-Red 12-bit unpacked |
| PixelFormat_SCF1WGWR12p             | Sparse Color Filter #1 White-Green-White-Red 12-bit packed   |
| PixelFormat_SCF1WGWR14              | Sparse Color Filter #1 White-Green-White-Red 14-bit unpacked |
| PixelFormat_SCF1WGWR16              | Sparse Color Filter #1 White-Green-White-Red 16-bit          |
| PixelFormat_SCF1WRWG8               | Sparse Color Filter #1 White-Red-White-Green 8-bit           |
| PixelFormat_SCF1WRWG10              | Sparse Color Filter #1 White-Red-White-Green 10-bit unpacked |
| PixelFormat_SCF1WRWG10p             | Sparse Color Filter #1 White-Red-White-Green 10-bit packed   |
| PixelFormat_SCF1WRWG12              | Sparse Color Filter #1 White-Red-White-Green 12-bit unpacked |
| PixelFormat_SCF1WRWG12p             | Sparse Color Filter #1 White-Red-White-Green 12-bit packed   |
| PixelFormat_SCF1WRWG14              | Sparse Color Filter #1 White-Red-White-Green 14-bit unpacked |
| PixelFormat_SCF1WRWG16              | Sparse Color Filter #1 White-Red-White-Green 16-bit          |
| PixelFormat_YCbCr8_CbYCr            | YCbCr 4:4:4 8-bit                                            |
| PixelFormat_YCbCr10_CbYCr           | YCbCr 4:4:4 10-bit unpacked                                  |
| PixelFormat_YCbCr10p_CbYCr          | YCbCr 4:4:4 10-bit packed                                    |
| PixelFormat_YCbCr12_CbYCr           | YCbCr 4:4:4 12-bit unpacked                                  |
| PixelFormat_YCbCr12p_CbYCr          | YCbCr 4:4:4 12-bit packed                                    |
| PixelFormat_YCbCr411_8_CbYYCrYY     | YCbCr 4:1:1 8-bit                                            |
| PixelFormat_YCbCr422_8_CbYCrY       | YCbCr 4:2:2 8-bit                                            |
| PixelFormat_YCbCr422_10             | YCbCr 4:2:2 10-bit unpacked                                  |
| PixelFormat_YCbCr422_10_CbYCrY      | YCbCr 4:2:2 10-bit unpacked                                  |
| PixelFormat_YCbCr422_10p            | YCbCr 4:2:2 10-bit packed                                    |
| PixelFormat_YCbCr422_10p_CbYCrY     | YCbCr 4:2:2 10-bit packed                                    |
| PixelFormat_YCbCr422_12             | YCbCr 4:2:2 12-bit unpacked                                  |
| PixelFormat_YCbCr422_12_CbYCrY      | YCbCr 4:2:2 12-bit unpacked                                  |
| PixelFormat_YCbCr422_12p            | YCbCr 4:2:2 12-bit packed                                    |
| PixelFormat_YCbCr422_12p_CbYCrY     | YCbCr 4:2:2 12-bit packed                                    |
| PixelFormat_YCbCr601_8_CbYCr        | YCbCr 4:4:4 8-bit BT.601                                     |
| PixelFormat_YCbCr601_10_CbYCr       | YCbCr 4:4:4 10-bit unpacked BT.601                           |
| PixelFormat_YCbCr601_10p_CbYCr      | YCbCr 4:4:4 10-bit packed BT.601                             |
| PixelFormat_YCbCr601_12_CbYCr       | YCbCr 4:4:4 12-bit unpacked BT.601                           |
| PixelFormat_YCbCr601_12p_CbYCr      | YCbCr 4:4:4 12-bit packed BT.601                             |
| PixelFormat_YCbCr601_411_8_CbYYCrYY | YCbCr 4:1:1 8-bit BT.601                                     |
| PixelFormat_YCbCr601_422_8          | YCbCr 4:2:2 8-bit BT.601                                     |
| PixelFormat_YCbCr601_422_8_CbYCrY   | YCbCr 4:2:2 8-bit BT.601                                     |
| PixelFormat_YCbCr601_422_10         | YCbCr 4:2:2 10-bit unpacked BT.601                           |
| PixelFormat_YCbCr601_422_10_CbYCrY  | YCbCr 4:2:2 10-bit unpacked BT.601                           |
| PixelFormat_YCbCr601_422_10p        | YCbCr 4:2:2 10-bit packed BT.601                             |
| PixelFormat_YCbCr601_422_10p_CbYCrY | YCbCr 4:2:2 10-bit packed BT.601                             |
| PixelFormat_YCbCr601_422_12         | YCbCr 4:2:2 12-bit unpacked BT.601                           |
| PixelFormat_YCbCr601_422_12_CbYCrY  | YCbCr 4:2:2 12-bit unpacked BT.601                           |
| PixelFormat_YCbCr601_422_12p        | YCbCr 4:2:2 12-bit packed BT.601                             |
| PixelFormat_YCbCr601_422_12p_CbYCrY | YCbCr 4:2:2 12-bit packed BT.601                             |
| PixelFormat_YCbCr709_8_CbYCr        | YCbCr 4:4:4 8-bit BT.709                                     |
| PixelFormat_YCbCr709_10_CbYCr       | YCbCr 4:4:4 10-bit unpacked BT.709                           |
| PixelFormat_YCbCr709_10p_CbYCr      | YCbCr 4:4:4 10-bit packed BT.709                             |
| PixelFormat_YCbCr709_12_CbYCr       | YCbCr 4:4:4 12-bit unpacked BT.709                           |
| PixelFormat_YCbCr709_12p_CbYCr      | YCbCr 4:4:4 12-bit packed BT.709                             |

## Enumerator

|                                     |                                                   |
|-------------------------------------|---------------------------------------------------|
| PixelFormat_YCbCr709_411_8_CbYYCrYY | YCbCr 4:1:1 8-bit BT.709                          |
| PixelFormat_YCbCr709_422_8          | YCbCr 4:2:2 8-bit BT.709                          |
| PixelFormat_YCbCr709_422_8_CbYCrY   | YCbCr 4:2:2 8-bit BT.709                          |
| PixelFormat_YCbCr709_422_10         | YCbCr 4:2:2 10-bit unpacked BT.709                |
| PixelFormat_YCbCr709_422_10_CbYCrY  | YCbCr 4:2:2 10-bit unpacked BT.709                |
| PixelFormat_YCbCr709_422_10p        | YCbCr 4:2:2 10-bit packed BT.709                  |
| PixelFormat_YCbCr709_422_10p_CbYCrY | YCbCr 4:2:2 10-bit packed BT.709                  |
| PixelFormat_YCbCr709_422_12         | YCbCr 4:2:2 12-bit unpacked BT.709                |
| PixelFormat_YCbCr709_422_12_CbYCrY  | YCbCr 4:2:2 12-bit unpacked BT.709                |
| PixelFormat_YCbCr709_422_12p        | YCbCr 4:2:2 12-bit packed BT.709                  |
| PixelFormat_YCbCr709_422_12p_CbYCrY | YCbCr 4:2:2 12-bit packed BT.709                  |
| PixelFormat_YUV8_UYV                | YUV 4:4:4 8-bit                                   |
| PixelFormat_YUV411_8_UYYVYY         | YUV 4:1:1 8-bit                                   |
| PixelFormat_YUV422_8                | YUV 4:2:2 8-bit                                   |
| PixelFormat_YUV422_8_UYVY           | YUV 4:2:2 8-bit                                   |
| PixelFormat_Polarized8              | Monochrome Polarized 8-bit                        |
| PixelFormat_Polarized10p            | Monochrome Polarized 10-bit packed                |
| PixelFormat_Polarized12p            | Monochrome Polarized 12-bit packed                |
| PixelFormat_Polarized16             | Monochrome Polarized 16-bit                       |
| PixelFormat_BayerRGPolarized8       | Polarized Bayer Red Green filter 8-bit            |
| PixelFormat_BayerRGPolarized10p     | Polarized Bayer Red Green filter 10-bit packed    |
| PixelFormat_BayerRGPolarized12p     | Polarized Bayer Red Green filter 12-bit packed    |
| PixelFormat_BayerRGPolarized16      | Polarized Bayer Red Green filter 16-bit           |
| PixelFormat_LLCMono8                | Lossless Compression Monochrome 8-bit             |
| PixelFormat_LLCBayerRG8             | Lossless Compression Bayer Red Green filter 8-bit |
| PixelFormat_JPEGMono8               | JPEG Monochrome 8-bit                             |
| PixelFormat_JPEGColor8              | JPEG Color 8-bit                                  |
| PixelFormat_Raw16                   | Raw 16 bit.                                       |
| PixelFormat_Raw8                    | Raw bit.                                          |
| PixelFormat_R12_Jpeg                | Red 12-bit JPEG.                                  |
| PixelFormat_GR12_Jpeg               | Green Red 12-bit JPEG.                            |
| PixelFormat_GB12_Jpeg               | Green Blue 12-bit JPEG.                           |
| PixelFormat_B12_Jpeg                | Blue 12-bit packed JPEG.                          |
| UNKNOWN_PIXELFORMAT                 |                                                   |
| NUM_PIXELFORMAT                     |                                                   |

## 6.1.1.127 \_spinPixelFormatInfoSelectorEnums

```
enum _spinPixelFormatInfoSelectorEnums
```

< Select the pixel format for which the information will be returned.

## Enumerator

|                                |                         |
|--------------------------------|-------------------------|
| PixelFormatInfoSelector_Mono1p | Monochrome 1-bit packed |
|--------------------------------|-------------------------|

## Enumerator

|                                      |                                       |
|--------------------------------------|---------------------------------------|
| PixelFormatInfoSelector_Mono2p       | Monochrome 2-bit packed               |
| PixelFormatInfoSelector_Mono4p       | Monochrome 4-bit packed               |
| PixelFormatInfoSelector_Mono8        | Monochrome 8-bit                      |
| PixelFormatInfoSelector_Mono8s       | Monochrome 8-bit signed               |
| PixelFormatInfoSelector_Mono10       | Monochrome 10-bit unpacked            |
| PixelFormatInfoSelector_Mono10p      | Monochrome 10-bit packed              |
| PixelFormatInfoSelector_Mono12       | Monochrome 12-bit unpacked            |
| PixelFormatInfoSelector_Mono12p      | Monochrome 12-bit packed              |
| PixelFormatInfoSelector_Mono14       | Monochrome 14-bit unpacked            |
| PixelFormatInfoSelector_Mono16       | Monochrome 16-bit                     |
| PixelFormatInfoSelector_Mono16s      | Monochrome 16-bit signed              |
| PixelFormatInfoSelector_Mono32f      | Monochrome 32-bit float               |
| PixelFormatInfoSelector_BayerBG8     | Bayer Blue-Green 8-bit                |
| PixelFormatInfoSelector_BayerBG10    | Bayer Blue-Green 10-bit unpacked      |
| PixelFormatInfoSelector_BayerBG10p   | Bayer Blue-Green 10-bit packed        |
| PixelFormatInfoSelector_BayerBG12    | Bayer Blue-Green 12-bit unpacked      |
| PixelFormatInfoSelector_BayerBG12p   | Bayer Blue-Green 12-bit packed        |
| PixelFormatInfoSelector_BayerBG16    | Bayer Blue-Green 16-bit               |
| PixelFormatInfoSelector_BayerGB8     | Bayer Green-Blue 8-bit                |
| PixelFormatInfoSelector_BayerGB10    | Bayer Green-Blue 10-bit unpacked      |
| PixelFormatInfoSelector_BayerGB10p   | Bayer Green-Blue 10-bit packed        |
| PixelFormatInfoSelector_BayerGB12    | Bayer Green-Blue 12-bit unpacked      |
| PixelFormatInfoSelector_BayerGB12p   | Bayer Green-Blue 12-bit packed        |
| PixelFormatInfoSelector_BayerGB16    | Bayer Green-Blue 16-bit               |
| PixelFormatInfoSelector_BayerGR8     | Bayer Green-Red 8-bit                 |
| PixelFormatInfoSelector_BayerGR10    | Bayer Green-Red 10-bit unpacked       |
| PixelFormatInfoSelector_BayerGR10p   | Bayer Green-Red 10-bit packed         |
| PixelFormatInfoSelector_BayerGR12    | Bayer Green-Red 12-bit unpacked       |
| PixelFormatInfoSelector_BayerGR12p   | Bayer Green-Red 12-bit packed         |
| PixelFormatInfoSelector_BayerGR16    | Bayer Green-Red 16-bit                |
| PixelFormatInfoSelector_BayerRG8     | Bayer Red-Green 8-bit                 |
| PixelFormatInfoSelector_BayerRG10    | Bayer Red-Green 10-bit unpacked       |
| PixelFormatInfoSelector_BayerRG10p   | Bayer Red-Green 10-bit packed         |
| PixelFormatInfoSelector_BayerRG12    | Bayer Red-Green 12-bit unpacked       |
| PixelFormatInfoSelector_BayerRG12p   | Bayer Red-Green 12-bit packed         |
| PixelFormatInfoSelector_BayerRG16    | Bayer Red-Green 16-bit                |
| PixelFormatInfoSelector_RGBa8        | Red-Green-Blue-alpha 8-bit            |
| PixelFormatInfoSelector_RGBa10       | Red-Green-Blue-alpha 10-bit unpacked  |
| PixelFormatInfoSelector_RGBa10p      | Red-Green-Blue-alpha 10-bit packed    |
| PixelFormatInfoSelector_RGBa12       | Red-Green-Blue-alpha 12-bit unpacked  |
| PixelFormatInfoSelector_RGBa12p      | Red-Green-Blue-alpha 12-bit packed    |
| PixelFormatInfoSelector_RGBa14       | Red-Green-Blue-alpha 14-bit unpacked  |
| PixelFormatInfoSelector_RGBa16       | Red-Green-Blue-alpha 16-bit           |
| PixelFormatInfoSelector_RGB8         | Red-Green-Blue 8-bit                  |
| PixelFormatInfoSelector_RGB8_Planar  | Red-Green-Blue 8-bit planar           |
| PixelFormatInfoSelector_RGB10        | Red-Green-Blue 10-bit unpacked        |
| PixelFormatInfoSelector_RGB10_Planar | Red-Green-Blue 10-bit unpacked planar |

## Enumerator

|                                               |                                          |
|-----------------------------------------------|------------------------------------------|
| PixelFormatInfoSelector_RGB10p                | Red-Green-Blue 10-bit packed             |
| PixelFormatInfoSelector_RGB10p32              | Red-Green-Blue 10-bit packed into 32-bit |
| PixelFormatInfoSelector_RGB12                 | Red-Green-Blue 12-bit unpacked           |
| PixelFormatInfoSelector_RGB12_Planar          | Red-Green-Blue 12-bit unpacked planar    |
| PixelFormatInfoSelector_RGB12p                | Red-Green-Blue 12-bit packed             |
| PixelFormatInfoSelector_RGB14                 | Red-Green-Blue 14-bit unpacked           |
| PixelFormatInfoSelector_RGB16                 | Red-Green-Blue 16-bit                    |
| PixelFormatInfoSelector_RGB16s                | Red-Green-Blue 16-bit signed             |
| PixelFormatInfoSelector_RGB32f                | Red-Green-Blue 32-bit float              |
| PixelFormatInfoSelector_RGB16_Planar          | Red-Green-Blue 16-bit planar             |
| PixelFormatInfoSelector_RGB565p               | Red-Green-Blue 5/6/5-bit packed          |
| PixelFormatInfoSelector_BGRa8                 | Blue-Green-Red-alpha 8-bit               |
| PixelFormatInfoSelector_BGRa10                | Blue-Green-Red-alpha 10-bit unpacked     |
| PixelFormatInfoSelector_BGRa10p               | Blue-Green-Red-alpha 10-bit packed       |
| PixelFormatInfoSelector_BGRa12                | Blue-Green-Red-alpha 12-bit unpacked     |
| PixelFormatInfoSelector_BGRa12p               | Blue-Green-Red-alpha 12-bit packed       |
| PixelFormatInfoSelector_BGRa14                | Blue-Green-Red-alpha 14-bit unpacked     |
| PixelFormatInfoSelector_BGRa16                | Blue-Green-Red-alpha 16-bit              |
| PixelFormatInfoSelector_RGBa32f               | Red-Green-Blue-alpha 32-bit float        |
| PixelFormatInfoSelector_BGR8                  | Blue-Green-Red 8-bit                     |
| PixelFormatInfoSelector_BGR10                 | Blue-Green-Red 10-bit unpacked           |
| PixelFormatInfoSelector_BGR10p                | Blue-Green-Red 10-bit packed             |
| PixelFormatInfoSelector_BGR12                 | Blue-Green-Red 12-bit unpacked           |
| PixelFormatInfoSelector_BGR12p                | Blue-Green-Red 12-bit packed             |
| PixelFormatInfoSelector_BGR14                 | Blue-Green-Red 14-bit unpacked           |
| PixelFormatInfoSelector_BGR16                 | Blue-Green-Red 16-bit                    |
| PixelFormatInfoSelector_BGR565p               | Blue-Green-Red 5/6/5-bit packed          |
| PixelFormatInfoSelector_R8                    | Red 8-bit                                |
| PixelFormatInfoSelector_R10                   | Red 10-bit                               |
| PixelFormatInfoSelector_R12                   | Red 12-bit                               |
| PixelFormatInfoSelector_R16                   | Red 16-bit                               |
| PixelFormatInfoSelector_G8                    | Green 8-bit                              |
| PixelFormatInfoSelector_G10                   | Green 10-bit                             |
| PixelFormatInfoSelector_G12                   | Green 12-bit                             |
| PixelFormatInfoSelector_G16                   | Green 16-bit                             |
| PixelFormatInfoSelector_B8                    | Blue 8-bit                               |
| PixelFormatInfoSelector_B10                   | Blue 10-bit                              |
| PixelFormatInfoSelector_B12                   | Blue 12-bit                              |
| PixelFormatInfoSelector_B16                   | Blue 16-bit                              |
| PixelFormatInfoSelector_Coord3D_ABC8          | 3D coordinate A-B-C 8-bit                |
| PixelFormatInfoSelector_Coord3D_ABC8_Planar   | 3D coordinate A-B-C 8-bit planar         |
| PixelFormatInfoSelector_Coord3D_ABC10p        | 3D coordinate A-B-C 10-bit packed        |
| PixelFormatInfoSelector_Coord3D_ABC10p_Planar | 3D coordinate A-B-C 10-bit packed planar |
| PixelFormatInfoSelector_Coord3D_ABC12p        | 3D coordinate A-B-C 12-bit packed        |
| PixelFormatInfoSelector_Coord3D_ABC12p_Planar | 3D coordinate A-B-C 12-bit packed planar |
| PixelFormatInfoSelector_Coord3D_ABC16         | 3D coordinate A-B-C 16-bit               |

## Enumerator

|                                               |                                                               |
|-----------------------------------------------|---------------------------------------------------------------|
| PixelFormatInfoSelector_Coord3D_ABC16_Planar  | 3D coordinate A-B-C 16-bit planar                             |
| PixelFormatInfoSelector_Coord3D_ABC32f        | 3D coordinate A-B-C 32-bit floating point                     |
| PixelFormatInfoSelector_Coord3D_ABC32f_Planar | 3D coordinate A-B-C 32-bit floating point planar              |
| PixelFormatInfoSelector_Coord3D_AC8           | 3D coordinate A-C 8-bit                                       |
| PixelFormatInfoSelector_Coord3D_AC8_Planar    | 3D coordinate A-C 8-bit planar                                |
| PixelFormatInfoSelector_Coord3D_AC10p         | 3D coordinate A-C 10-bit packed                               |
| PixelFormatInfoSelector_Coord3D_AC10p_Planar  | 3D coordinate A-C 10-bit packed planar                        |
| PixelFormatInfoSelector_Coord3D_AC12p         | 3D coordinate A-C 12-bit packed                               |
| PixelFormatInfoSelector_Coord3D_AC12p_Planar  | 3D coordinate A-C 12-bit packed planar                        |
| PixelFormatInfoSelector_Coord3D_AC16          | 3D coordinate A-C 16-bit                                      |
| PixelFormatInfoSelector_Coord3D_AC16_Planar   | 3D coordinate A-C 16-bit planar                               |
| PixelFormatInfoSelector_Coord3D_AC32f         | 3D coordinate A-C 32-bit floating point                       |
| PixelFormatInfoSelector_Coord3D_AC32f_Planar  | 3D coordinate A-C 32-bit floating point planar                |
| PixelFormatInfoSelector_Coord3D_A8            | 3D coordinate A 8-bit                                         |
| PixelFormatInfoSelector_Coord3D_A10p          | 3D coordinate A 10-bit packed                                 |
| PixelFormatInfoSelector_Coord3D_A12p          | 3D coordinate A 12-bit packed                                 |
| PixelFormatInfoSelector_Coord3D_A16           | 3D coordinate A 16-bit                                        |
| PixelFormatInfoSelector_Coord3D_A32f          | 3D coordinate A 32-bit floating point                         |
| PixelFormatInfoSelector_Coord3D_B8            | 3D coordinate B 8-bit                                         |
| PixelFormatInfoSelector_Coord3D_B10p          | 3D coordinate B 10-bit packed                                 |
| PixelFormatInfoSelector_Coord3D_B12p          | 3D coordinate B 12-bit packed                                 |
| PixelFormatInfoSelector_Coord3D_B16           | 3D coordinate B 16-bit                                        |
| PixelFormatInfoSelector_Coord3D_B32f          | 3D coordinate B 32-bit floating point                         |
| PixelFormatInfoSelector_Coord3D_C8            | 3D coordinate C 8-bit                                         |
| PixelFormatInfoSelector_Coord3D_C10p          | 3D coordinate C 10-bit packed                                 |
| PixelFormatInfoSelector_Coord3D_C12p          | 3D coordinate C 12-bit packed                                 |
| PixelFormatInfoSelector_Coord3D_C16           | 3D coordinate C 16-bit                                        |
| PixelFormatInfoSelector_Coord3D_C32f          | 3D coordinate C 32-bit floating point                         |
| PixelFormatInfoSelector_Confidence1           | Confidence 1-bit unpacked                                     |
| PixelFormatInfoSelector_Confidence1p          | Confidence 1-bit packed                                       |
| PixelFormatInfoSelector_Confidence8           | Confidence 8-bit                                              |
| PixelFormatInfoSelector_Confidence16          | Confidence 16-bit                                             |
| PixelFormatInfoSelector_Confidence32f         | Confidence 32-bit floating point                              |
| PixelFormatInfoSelector_BiColorBGRG8          | Bi-color Blue/Green - Red/Green 8-bit                         |
| PixelFormatInfoSelector_BiColorBGRG10         | Bi-color Blue/Green - Red/Green 10-bit unpacked               |
| PixelFormatInfoSelector_BiColorBGRG10p        | Bi-color Blue/Green - Red/Green 10-bit packed                 |
| PixelFormatInfoSelector_BiColorBGRG12         | Bi-color Blue/Green - Red/Green 12-bit unpacked               |
| PixelFormatInfoSelector_BiColorBGRG12p        | Bi-color Blue/Green - Red/Green 12-bit packed                 |
| PixelFormatInfoSelector_BiColorRGBG8          | Bi-color Red/Green - Blue/Green 8-bit                         |
| PixelFormatInfoSelector_BiColorRGBG10         | Bi-color Red/Green - Blue/Green 10-bit unpacked               |
| PixelFormatInfoSelector_BiColorRGBG10p        | Bi-color Red/Green - Blue/Green 10-bit packed                 |
| PixelFormatInfoSelector_BiColorRGBG12         | Bi-color Red/Green - Blue/Green 12-bit unpacked               |
| PixelFormatInfoSelector_BiColorRGBG12p        | Bi-color Red/Green - Blue/Green 12-bit packed                 |
| PixelFormatInfoSelector_SCF1WBWG8             | Sparse Color Filter #1 White-Blue-White-Green 8-bit           |
| PixelFormatInfoSelector_SCF1WBWG10            | Sparse Color Filter #1 White-Blue-White-Green 10-bit unpacked |

## Enumerator

|                                        |                                                               |
|----------------------------------------|---------------------------------------------------------------|
| PixelFormatInfoSelector_SCF1WBWG10p    | Sparse Color Filter #1 White-Blue-White-Green 10-bit packed   |
| PixelFormatInfoSelector_SCF1WBWG12     | Sparse Color Filter #1 White-Blue-White-Green 12-bit unpacked |
| PixelFormatInfoSelector_SCF1WBWG12p    | Sparse Color Filter #1 White-Blue-White-Green 12-bit packed   |
| PixelFormatInfoSelector_SCF1WBWG14     | Sparse Color Filter #1 White-Blue-White-Green 14-bit unpacked |
| PixelFormatInfoSelector_SCF1WBWG16     | Sparse Color Filter #1 White-Blue-White-Green 16-bit unpacked |
| PixelFormatInfoSelector_SCF1WGWB8      | Sparse Color Filter #1 White-Green-White-Blue 8-bit           |
| PixelFormatInfoSelector_SCF1WGWB10     | Sparse Color Filter #1 White-Green-White-Blue 10-bit unpacked |
| PixelFormatInfoSelector_SCF1WGWB10p    | Sparse Color Filter #1 White-Green-White-Blue 10-bit packed   |
| PixelFormatInfoSelector_SCF1WGWB12     | Sparse Color Filter #1 White-Green-White-Blue 12-bit unpacked |
| PixelFormatInfoSelector_SCF1WGWB12p    | Sparse Color Filter #1 White-Green-White-Blue 12-bit packed   |
| PixelFormatInfoSelector_SCF1WGWB14     | Sparse Color Filter #1 White-Green-White-Blue 14-bit unpacked |
| PixelFormatInfoSelector_SCF1WGWB16     | Sparse Color Filter #1 White-Green-White-Blue 16-bit          |
| PixelFormatInfoSelector_SCF1WGWR8      | Sparse Color Filter #1 White-Green-White-Red 8-bit            |
| PixelFormatInfoSelector_SCF1WGWR10     | Sparse Color Filter #1 White-Green-White-Red 10-bit unpacked  |
| PixelFormatInfoSelector_SCF1WGWR10p    | Sparse Color Filter #1 White-Green-White-Red 10-bit packed    |
| PixelFormatInfoSelector_SCF1WGWR12     | Sparse Color Filter #1 White-Green-White-Red 12-bit unpacked  |
| PixelFormatInfoSelector_SCF1WGWR12p    | Sparse Color Filter #1 White-Green-White-Red 12-bit packed    |
| PixelFormatInfoSelector_SCF1WGWR14     | Sparse Color Filter #1 White-Green-White-Red 14-bit unpacked  |
| PixelFormatInfoSelector_SCF1WGWR16     | Sparse Color Filter #1 White-Green-White-Red 16-bit           |
| PixelFormatInfoSelector_SCF1WRWG8      | Sparse Color Filter #1 White-Red-White-Green 8-bit            |
| PixelFormatInfoSelector_SCF1WRWG10     | Sparse Color Filter #1 White-Red-White-Green 10-bit unpacked  |
| PixelFormatInfoSelector_SCF1WRWG10p    | Sparse Color Filter #1 White-Red-White-Green 10-bit packed    |
| PixelFormatInfoSelector_SCF1WRWG12     | Sparse Color Filter #1 White-Red-White-Green 12-bit unpacked  |
| PixelFormatInfoSelector_SCF1WRWG12p    | Sparse Color Filter #1 White-Red-White-Green 12-bit packed    |
| PixelFormatInfoSelector_SCF1WRWG14     | Sparse Color Filter #1 White-Red-White-Green 14-bit unpacked  |
| PixelFormatInfoSelector_SCF1WRWG16     | Sparse Color Filter #1 White-Red-White-Green 16-bit           |
| PixelFormatInfoSelector_YCbCr8         | YCbCr 4:4:4 8-bit                                             |
| PixelFormatInfoSelector_YCbCr8_CbYCr   | YCbCr 4:4:4 8-bit                                             |
| PixelFormatInfoSelector_YCbCr10_CbYCr  | YCbCr 4:4:4 10-bit unpacked                                   |
| PixelFormatInfoSelector_YCbCr10p_CbYCr | YCbCr 4:4:4 10-bit packed                                     |

## Enumerator

|                                                      |                                    |
|------------------------------------------------------|------------------------------------|
| PixelFormatInfoSelector_YCbCr12_CbYCr                | YCbCr 4:4:4 12-bit unpacked        |
| PixelFormatInfoSelector_YCbCr12p_CbYCr               | YCbCr 4:4:4 12-bit packed          |
| PixelFormatInfoSelector_YCbCr411_8                   | YCbCr 4:1:1 8-bit                  |
| PixelFormatInfoSelector_YCbCr411_8_CbYYCrYY          | YCbCr 4:1:1 8-bit                  |
| PixelFormatInfoSelector_YCbCr422_8                   | YCbCr 4:2:2 8-bit                  |
| PixelFormatInfoSelector_YCbCr422_8_CbYCrY            | YCbCr 4:2:2 8-bit                  |
| PixelFormatInfoSelector_YCbCr422_10                  | YCbCr 4:2:2 10-bit unpacked        |
| PixelFormatInfoSelector_YCbCr422_10_CbYCrY           | YCbCr 4:2:2 10-bit unpacked        |
| PixelFormatInfoSelector_YCbCr422_10p                 | YCbCr 4:2:2 10-bit packed          |
| PixelFormatInfoSelector_YCbCr422_10p_CbYCrY          | YCbCr 4:2:2 10-bit packed          |
| PixelFormatInfoSelector_YCbCr422_12                  | YCbCr 4:2:2 12-bit unpacked        |
| PixelFormatInfoSelector_YCbCr422_12_CbYCrY           | YCbCr 4:2:2 12-bit unpacked        |
| PixelFormatInfoSelector_YCbCr422_12p                 | YCbCr 4:2:2 12-bit packed          |
| PixelFormatInfoSelector_YCbCr422_12p_CbYCrY          | YCbCr 4:2:2 12-bit packed          |
| PixelFormatInfoSelector_YCbCr601_8_CbYCr             | YCbCr 4:4:4 8-bit BT.601           |
| PixelFormatInfoSelector_YCbCr601_10_CbYCr            | YCbCr 4:4:4 10-bit unpacked BT.601 |
| PixelFormatInfoSelector_YCbCr601_10p_CbYCr           | YCbCr 4:4:4 10-bit packed BT.601   |
| PixelFormatInfoSelector_YCbCr601_12_CbYCr            | YCbCr 4:4:4 12-bit unpacked BT.601 |
| PixelFormatInfoSelector_YCbCr601_12p_CbYCr           | YCbCr 4:4:4 12-bit packed BT.601   |
| PixelFormatInfoSelector_YCbCr601_411_8_CbYY↔<br>CrYY | YCbCr 4:1:1 8-bit BT.601           |
| PixelFormatInfoSelector_YCbCr601_422_8               | YCbCr 4:2:2 8-bit BT.601           |
| PixelFormatInfoSelector_YCbCr601_422_8_CbYCrY        | YCbCr 4:2:2 8-bit BT.601           |
| PixelFormatInfoSelector_YCbCr601_422_10              | YCbCr 4:2:2 10-bit unpacked BT.601 |
| PixelFormatInfoSelector_YCbCr601_422_10_CbY↔<br>CrY  | YCbCr 4:2:2 10-bit unpacked BT.601 |
| PixelFormatInfoSelector_YCbCr601_422_10p             | YCbCr 4:2:2 10-bit packed BT.601   |
| PixelFormatInfoSelector_YCbCr601_422_10p_Cb↔<br>YCrY | YCbCr 4:2:2 10-bit packed BT.601   |
| PixelFormatInfoSelector_YCbCr601_422_12              | YCbCr 4:2:2 12-bit unpacked BT.601 |
| PixelFormatInfoSelector_YCbCr601_422_12_CbY↔<br>CrY  | YCbCr 4:2:2 12-bit unpacked BT.601 |
| PixelFormatInfoSelector_YCbCr601_422_12p             | YCbCr 4:2:2 12-bit packed BT.601   |
| PixelFormatInfoSelector_YCbCr601_422_12p_Cb↔<br>YCrY | YCbCr 4:2:2 12-bit packed BT.601   |
| PixelFormatInfoSelector_YCbCr709_8_CbYCr             | YCbCr 4:4:4 8-bit BT.709           |
| PixelFormatInfoSelector_YCbCr709_10_CbYCr            | YCbCr 4:4:4 10-bit unpacked BT.709 |
| PixelFormatInfoSelector_YCbCr709_10p_CbYCr           | YCbCr 4:4:4 10-bit packed BT.709   |
| PixelFormatInfoSelector_YCbCr709_12_CbYCr            | YCbCr 4:4:4 12-bit unpacked BT.709 |
| PixelFormatInfoSelector_YCbCr709_12p_CbYCr           | YCbCr 4:4:4 12-bit packed BT.709   |
| PixelFormatInfoSelector_YCbCr709_411_8_CbYY↔<br>CrYY | YCbCr 4:1:1 8-bit BT.709           |
| PixelFormatInfoSelector_YCbCr709_422_8               | YCbCr 4:2:2 8-bit BT.709           |
| PixelFormatInfoSelector_YCbCr709_422_8_CbYCrY        | YCbCr 4:2:2 8-bit BT.709           |
| PixelFormatInfoSelector_YCbCr709_422_10              | YCbCr 4:2:2 10-bit unpacked BT.709 |
| PixelFormatInfoSelector_YCbCr709_422_10_CbY↔<br>CrY  | YCbCr 4:2:2 10-bit unpacked BT.709 |
| PixelFormatInfoSelector_YCbCr709_422_10p             | YCbCr 4:2:2 10-bit packed BT.709   |



## Enumerator

|                                                  |                                                   |
|--------------------------------------------------|---------------------------------------------------|
| PixelFormatInfoSelector_YCbCr709_422_10p_Cb↔YCrY | YCbCr 4:2:2 10-bit packed BT.709                  |
| PixelFormatInfoSelector_YCbCr709_422_12          | YCbCr 4:2:2 12-bit unpacked BT.709                |
| PixelFormatInfoSelector_YCbCr709_422_12_CbY↔CrY  | YCbCr 4:2:2 12-bit unpacked BT.709                |
| PixelFormatInfoSelector_YCbCr709_422_12p         | YCbCr 4:2:2 12-bit packed BT.709                  |
| PixelFormatInfoSelector_YCbCr709_422_12p_Cb↔YCrY | YCbCr 4:2:2 12-bit packed BT.709                  |
| PixelFormatInfoSelector_YUV8_UYV                 | YUV 4:4:4 8-bit                                   |
| PixelFormatInfoSelector_YUV411_8_UYYYVYY         | YUV 4:1:1 8-bit                                   |
| PixelFormatInfoSelector_YUV422_8                 | YUV 4:2:2 8-bit                                   |
| PixelFormatInfoSelector_YUV422_8_UYVY            | YUV 4:2:2 8-bit                                   |
| PixelFormatInfoSelector_Polarized8               | Monochrome Polarized 8-bit                        |
| PixelFormatInfoSelector_Polarized10p             | Monochrome Polarized 10-bit packed                |
| PixelFormatInfoSelector_Polarized12p             | Monochrome Polarized 12-bit packed                |
| PixelFormatInfoSelector_Polarized16              | Monochrome Polarized 16-bit                       |
| PixelFormatInfoSelector_BayerRGPolarized8        | Polarized Bayer Red Green filter 8-bit            |
| PixelFormatInfoSelector_BayerRGPolarized10p      | Polarized Bayer Red Green filter 10-bit packed    |
| PixelFormatInfoSelector_BayerRGPolarized12p      | Polarized Bayer Red Green filter 12-bit packed    |
| PixelFormatInfoSelector_BayerRGPolarized16       | Polarized Bayer Red Green filter 16-bit           |
| PixelFormatInfoSelector_LLCMono8                 | Lossless Compression Monochrome 8-bit             |
| PixelFormatInfoSelector_LLCBayerRG8              | Lossless Compression Bayer Red Green filter 8-bit |
| PixelFormatInfoSelector_JPEGMono8                | JPEG Monochrome 8-bit                             |
| PixelFormatInfoSelector_JPEGColor8               | JPEG Color 8-bit                                  |
| NUM_PIXELFORMATINFOSELECTOR                      |                                                   |

## 6.1.1.128 \_spinPixelFormatSizeEnums

```
enum _spinPixelFormatSizeEnums
```

< Total size in bits of a pixel of the image.

## Enumerator

|                       |                    |
|-----------------------|--------------------|
| PixelFormatSize_Bpp1  | 1 bit per pixel.   |
| PixelFormatSize_Bpp2  | 2 bits per pixel.  |
| PixelFormatSize_Bpp4  | 4 bits per pixel.  |
| PixelFormatSize_Bpp8  | 8 bits per pixel.  |
| PixelFormatSize_Bpp10 | 10 bits per pixel. |
| PixelFormatSize_Bpp12 | 12 bits per pixel. |
| PixelFormatSize_Bpp14 | 14 bits per pixel. |
| PixelFormatSize_Bpp16 | 16 bits per pixel. |
| PixelFormatSize_Bpp20 | 20 bits per pixel. |
| PixelFormatSize_Bpp24 | 24 bits per pixel. |
| PixelFormatSize_Bpp30 | 30 bits per pixel. |
| PixelFormatSize_Bpp32 | 32 bits per pixel. |

**Enumerator**

|                 |                    |
|-----------------|--------------------|
| PixelSize_Bpp36 | 36 bits per pixel. |
| PixelSize_Bpp48 | 48 bits per pixel. |
| PixelSize_Bpp64 | 64 bits per pixel. |
| PixelSize_Bpp96 | 96 bits per pixel. |
| NUM_PIXELSIZE   |                    |

**6.1.1.129 \_spinRegionDestinationEnums**

```
enum _spinRegionDestinationEnums
```

< Control the destination of the selected region.

**Enumerator**

|                           |                                                     |
|---------------------------|-----------------------------------------------------|
| RegionDestination_Stream0 | The destination of the region is the data stream 0. |
| RegionDestination_Stream1 | The destination of the region is the data stream 1. |
| RegionDestination_Stream2 | The destination of the region is the data stream 2. |
| NUM_REGIONDESTINATION     |                                                     |

**6.1.1.130 \_spinRegionModeEnums**

```
enum _spinRegionModeEnums
```

< Controls if the selected Region of interest is active and streaming.

**Enumerator**

|                |                                  |
|----------------|----------------------------------|
| RegionMode_Off | Disable the usage of the Region. |
| RegionMode_On  | Enable the usage of the Region.  |
| NUM_REGIONMODE |                                  |

**6.1.1.131 \_spinRegionSelectorEnums**

```
enum _spinRegionSelectorEnums
```

< Selects the Region of interest to control. The RegionSelector feature allows devices that are able to extract multiple regions out of an image, to configure the features of those individual regions independently.

## Enumerator

|                        |                                                                  |
|------------------------|------------------------------------------------------------------|
| RegionSelector_Region0 | Selected feature will control the region 0.                      |
| RegionSelector_Region1 | Selected feature will control the region 1.                      |
| RegionSelector_Region2 | Selected feature will control the region 2.                      |
| RegionSelector_All     | Selected features will control all the regions at the same time. |
| NUM_REGIONSELECTOR     |                                                                  |

## 6.1.1.132 \_spinRgbTransformLightSourceEnums

enum `_spinRgbTransformLightSourceEnums`

< Used to select from a set of RGBtoRGB transform matrices calibrated for different light sources. Selecting a value also sets the white balance ratios (BalanceRatioRed and BalanceRatioBlue), but those can be overwritten through manual or auto white balance.

## Enumerator

|                                              |                                                                                                                          |
|----------------------------------------------|--------------------------------------------------------------------------------------------------------------------------|
| RgbTransformLightSource_General              | Uses a matrix calibrated for a wide range of light sources.                                                              |
| RgbTransformLightSource_Tungsten2800K        | Uses a matrix optimized for tungsten/incandescent light with color temperature 2800K.                                    |
| RgbTransformLightSource_WarmFluorescent3000K | Uses a matrix optimized for a typical warm fluorescent light with color temperature 3000K.                               |
| RgbTransformLightSource_CoolFluorescent4000K | Uses a matrix optimized for a typical cool fluorescent light with color temperature 4000K.                               |
| RgbTransformLightSource_Daylight5000K        | Uses a matrix optimized for noon Daylight with color temperature 5000K.                                                  |
| RgbTransformLightSource_Cloudy6500K          | Uses a matrix optimized for a cloudy sky with color temperature 6500K.                                                   |
| RgbTransformLightSource_Shade8000K           | Uses a matrix optimized for shade with color temperature 8000K.                                                          |
| RgbTransformLightSource_Custom               | Uses a custom matrix set by the user through the ColorTransformationValueSelector and ColorTransformationValue controls. |
| NUM_RGBTRANSFORMLIGHTSOURCE                  |                                                                                                                          |

## 6.1.1.133 \_spinScan3dCoordinateReferenceSelectorEnums

enum `_spinScan3dCoordinateReferenceSelectorEnums`

< Sets the index to read a coordinate system reference value defining the transform of a point from the current (Anchor or Transformed) system to the reference system.

**Enumerator**

|                                                |                         |
|------------------------------------------------|-------------------------|
| Scan3dCoordinateReferenceSelector_RotationX    | Rotation around X axis. |
| Scan3dCoordinateReferenceSelector_RotationY    | Rotation around Y axis. |
| Scan3dCoordinateReferenceSelector_RotationZ    | Rotation around Z axis. |
| Scan3dCoordinateReferenceSelector_TranslationX | X axis translation.     |
| Scan3dCoordinateReferenceSelector_TranslationY | Y axis translation.     |
| Scan3dCoordinateReferenceSelector_TranslationZ | Z axis translation.     |
| NUM_SCAN3DCOORDINATEREFERENCESELECTOR          |                         |

**6.1.1.134 \_spinScan3dCoordinateSelectorEnums**

```
enum _spinScan3dCoordinateSelectorEnums
```

< Selects the individual coordinates in the vectors for 3D information/transformation.

**Enumerator**

|                                      |                                   |
|--------------------------------------|-----------------------------------|
| Scan3dCoordinateSelector_CoordinateA | The first (X or Theta) coordinate |
| Scan3dCoordinateSelector_CoordinateB | The second (Y or Phi) coordinate  |
| Scan3dCoordinateSelector_CoordinateC | The third (Z or Rho) coordinate.  |
| NUM_SCAN3DCOORDINATESELECTOR         |                                   |

**6.1.1.135 \_spinScan3dCoordinateSystemEnums**

```
enum _spinScan3dCoordinateSystemEnums
```

< Specifies the Coordinate system to use for the device.

**Enumerator**

|                                    |                                                     |
|------------------------------------|-----------------------------------------------------|
| Scan3dCoordinateSystem_Cartesian   | Default value. 3-axis orthogonal, right-hand X-Y-Z. |
| Scan3dCoordinateSystem_Spherical   | A Theta-Phi-Rho coordinate system.                  |
| Scan3dCoordinateSystem_Cylindrical | A Theta-Y-Rho coordinate system.                    |
| NUM_SCAN3DCOORDINATESYSTEM         |                                                     |

**6.1.1.136 \_spinScan3dCoordinateSystemReferenceEnums**

```
enum _spinScan3dCoordinateSystemReferenceEnums
```

< Defines coordinate system reference location.

## Enumerator

|                                             |                                                                                                                                               |
|---------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| Scan3dCoordinateSystemReference_Anchor      | Default value. Original fixed reference. The coordinate system fixed relative the camera reference point marker is used.                      |
| Scan3dCoordinateSystemReference_Transformed | Transformed reference system. The transformed coordinate system is used according to the definition in the rotation and translation matrices. |
| NUM_SCAN3DCOORDINATESYSTEMREFERENCE         |                                                                                                                                               |

## 6.1.1.137 \_spinScan3dCoordinateTransformSelectorEnums

enum `_spinScan3dCoordinateTransformSelectorEnums`

< Sets the index to read/write a coordinate transform value.

## Enumerator

|                                                |                           |
|------------------------------------------------|---------------------------|
| Scan3dCoordinateTransformSelector_RotationX    | Rotation around X axis.   |
| Scan3dCoordinateTransformSelector_RotationY    | Rotation around Y axis.   |
| Scan3dCoordinateTransformSelector_RotationZ    | Rotation around Z axis.   |
| Scan3dCoordinateTransformSelector_TranslationX | Translation along X axis. |
| Scan3dCoordinateTransformSelector_TranslationY | Translation along Y axis. |
| Scan3dCoordinateTransformSelector_TranslationZ | Translation along Z axis. |
| NUM_SCAN3DCOORDINATETRANSFORMSELECTOR          |                           |

## 6.1.1.138 \_spinScan3dDistanceUnitEnums

enum `_spinScan3dDistanceUnitEnums`

< Specifies the unit used when delivering calibrated distance data.

## Enumerator

|                               |                                                    |
|-------------------------------|----------------------------------------------------|
| Scan3dDistanceUnit_Millimeter | Distance values are in millimeter units (default). |
| Scan3dDistanceUnit_Inch       | Distance values are in inch units.                 |
| NUM_SCAN3DDISTANCEUNIT        |                                                    |

## 6.1.1.139 \_spinScan3dOutputModeEnums

enum `_spinScan3dOutputModeEnums`

< Controls the Calibration and data organization of the device, naming the coordinates transmitted.

#### Enumerator

|                                           |                                                                                                                                                                                                                                                                                                             |
|-------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Scan3dOutputMode_UncalibratedC            | Uncalibrated 2.5D Depth map. The distance data does not represent a physical unit and may be non-linear. The data is a 2.5D range map only.                                                                                                                                                                 |
| Scan3dOutputMode_CalibratedABC_Grid       | 3 Coordinates in grid organization. The full 3 coordinate data with the grid array organization from the sensor kept.                                                                                                                                                                                       |
| Scan3dOutputMode_CalibratedABC_PointCloud | 3 Coordinates without organization. The full 3 coordinate data without any organization of data points. Typically only valid points transmitted giving varying image size.                                                                                                                                  |
| Scan3dOutputMode_CalibratedAC             | 2 Coordinates with fixed B sampling. The data is sent as a A and C coordinates (X,Z or Theta,Rho). The B (Y) axis uses the scale and offset parameters for the B axis.                                                                                                                                      |
| Scan3dOutputMode_CalibratedAC_Linescan    | 2 Coordinates with varying sampling. The data is sent as a A and C coordinates (X,Z or Theta,Rho). The B (Y) axis comes from the encoder chunk value.                                                                                                                                                       |
| Scan3dOutputMode_CalibratedC              | Calibrated 2.5D Depth map. The distance data is expressed in the chosen distance unit. The data is a 2.5D range map. No information on X-Y axes available.                                                                                                                                                  |
| Scan3dOutputMode_CalibratedC_Linescan     | Depth Map with varying B sampling. The distance data is expressed in the chosen distance unit. The data is a 2.5D range map. The B (Y) axis comes from the encoder chunk value.                                                                                                                             |
| Scan3dOutputMode_RectifiedC               | Rectified 2.5D Depth map. The distance data has been rectified to a uniform sampling pattern in the X and Y direction. The data is a 2.5D range map only. If a complete 3D point cloud is rectified but transmitted as explicit coordinates it should be transmitted as one of the "CalibratedABC" formats. |
| Scan3dOutputMode_RectifiedC_Linescan      | Rectified 2.5D Depth map with varying B sampling. The data is sent as rectified 1D profiles using Coord3D_C pixels. The B (Y) axis comes from the encoder chunk value.                                                                                                                                      |
| Scan3dOutputMode_DisparityC               | Disparity 2.5D Depth map. The distance is inversely proportional to the pixel (disparity) value.                                                                                                                                                                                                            |
| Scan3dOutputMode_DisparityC_Linescan      | Disparity 2.5D Depth map with varying B sampling. The distance is inversely proportional to the pixel (disparity) value. The B (Y) axis comes from the encoder chunk value.                                                                                                                                 |
| NUM_SCAN3DOUTPUTMODE                      |                                                                                                                                                                                                                                                                                                             |

#### 6.1.1.140 \_spinSensorDigitizationTapsEnums

enum `_spinSensorDigitizationTapsEnums`

< Number of digitized samples outputted simultaneously by the camera A/D conversion stage.

#### Enumerator

|                            |        |
|----------------------------|--------|
| SensorDigitizationTaps_One | 1 tap. |
|----------------------------|--------|

## Enumerator

|                              |          |
|------------------------------|----------|
| SensorDigitizationTaps_Two   | 2 taps.  |
| SensorDigitizationTaps_Three | 3 taps.  |
| SensorDigitizationTaps_Four  | 4 taps.  |
| SensorDigitizationTaps_Eight | 8 taps.  |
| SensorDigitizationTaps_Ten   | 10 taps. |
| NUM_SENSORDIGITIZATIONTAPS   |          |

## 6.1.1.141 \_spinSensorShutterModeEnums

enum `_spinSensorShutterModeEnums`

< Sets the shutter mode of the device.

## Enumerator

|                               |                                                                                                                                                                    |
|-------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SensorShutterMode_Global      | The shutter opens and closes at the same time for all pixels. All the pixels are exposed for the same length of time at the same time.                             |
| SensorShutterMode_Rolling     | The shutter opens and closes sequentially for groups (typically lines) of pixels. All the pixels are exposed for the same length of time but not at the same time. |
| SensorShutterMode_GlobalReset | The shutter opens at the same time for all pixels but ends in a sequential manner. The pixels are exposed for different lengths of time.                           |
| NUM_SENSORSHUTTERMODE         |                                                                                                                                                                    |

## 6.1.1.142 \_spinSensorTapsEnums

enum `_spinSensorTapsEnums`

< Number of taps of the camera sensor.

## Enumerator

|                  |          |
|------------------|----------|
| SensorTaps_One   | 1 tap.   |
| SensorTaps_Two   | 2 taps.  |
| SensorTaps_Three | 3 taps.  |
| SensorTaps_Four  | 4 taps.  |
| SensorTaps_Eight | 8 taps.  |
| SensorTaps_Ten   | 10 taps. |
| NUM_SENSORTAPS   |          |

**6.1.1.143 \_spinSequencerConfigurationModeEnums**

```
enum _spinSequencerConfigurationModeEnums
```

< Controls whether or not a sequencer is in configuration mode.

Enumerator

|                                |  |
|--------------------------------|--|
| SequencerConfigurationMode_Off |  |
| SequencerConfigurationMode_On  |  |
| NUM_SEQUENCERCONFIGURATIONMODE |  |

**6.1.1.144 \_spinSequencerConfigurationValidEnums**

```
enum _spinSequencerConfigurationValidEnums
```

< Display whether the current sequencer configuration is valid to run.

Enumerator

|                                 |  |
|---------------------------------|--|
| SequencerConfigurationValid_No  |  |
| SequencerConfigurationValid_Yes |  |
| NUM_SEQUENCERCONFIGURATIONVALID |  |

**6.1.1.145 \_spinSequencerModeEnums**

```
enum _spinSequencerModeEnums
```

< Controls whether or not a sequencer is active.

Enumerator

|                   |  |
|-------------------|--|
| SequencerMode_Off |  |
| SequencerMode_On  |  |
| NUM_SEQUENCERMODE |  |

**6.1.1.146 \_spinSequencerSetValidEnums**

```
enum _spinSequencerSetValidEnums
```

< Displays whether the currently selected sequencer set's register contents are valid to use.



## Enumerator

|                       |  |
|-----------------------|--|
| SequencerSetValid_No  |  |
| SequencerSetValid_Yes |  |
| NUM_SEQUENCERSETVALID |  |

**6.1.1.147 \_spinSequencerTriggerActivationEnums**

```
enum _spinSequencerTriggerActivationEnums
```

< Specifies the activation mode of the sequencer trigger.

## Enumerator

|                                        |  |
|----------------------------------------|--|
| SequencerTriggerActivation_RisingEdge  |  |
| SequencerTriggerActivation_FallingEdge |  |
| SequencerTriggerActivation_AnyEdge     |  |
| SequencerTriggerActivation_LevelHigh   |  |
| SequencerTriggerActivation_LevelLow    |  |
| NUM_SEQUENCERTRIGGERACTIVATION         |  |

**6.1.1.148 \_spinSequencerTriggerSourceEnums**

```
enum _spinSequencerTriggerSourceEnums
```

< Specifies the internal signal or physical input line to use as the sequencer trigger source.

## Enumerator

|                                   |  |
|-----------------------------------|--|
| SequencerTriggerSource_Off        |  |
| SequencerTriggerSource_FrameStart |  |
| NUM_SEQUENCERTRIGGERSOURCE        |  |

**6.1.1.149 \_spinSerialPortBaudRateEnums**

```
enum _spinSerialPortBaudRateEnums
```

< This feature controls the baud rate used by the selected serial port.

## Enumerator

|                               |  |
|-------------------------------|--|
| SerialPortBaudRate_Baud300    |  |
| SerialPortBaudRate_Baud600    |  |
| SerialPortBaudRate_Baud1200   |  |
| SerialPortBaudRate_Baud2400   |  |
| SerialPortBaudRate_Baud4800   |  |
| SerialPortBaudRate_Baud9600   |  |
| SerialPortBaudRate_Baud14400  |  |
| SerialPortBaudRate_Baud19200  |  |
| SerialPortBaudRate_Baud38400  |  |
| SerialPortBaudRate_Baud57600  |  |
| SerialPortBaudRate_Baud115200 |  |
| SerialPortBaudRate_Baud230400 |  |
| SerialPortBaudRate_Baud460800 |  |
| SerialPortBaudRate_Baud921600 |  |
| NUM_SERIALPORTBAUDRATE        |  |

**6.1.1.150 \_spinSerialPortParityEnums**

```
enum _spinSerialPortParityEnums
```

< This feature controls the parity used by the selected serial port.

## Enumerator

|                        |  |
|------------------------|--|
| SerialPortParity_None  |  |
| SerialPortParity_Odd   |  |
| SerialPortParity_Even  |  |
| SerialPortParity_Mark  |  |
| SerialPortParity_Space |  |
| NUM_SERIALPORTPARITY   |  |

**6.1.1.151 \_spinSerialPortSelectorEnums**

```
enum _spinSerialPortSelectorEnums
```

< Selects which serial port of the device to control.

## Enumerator

|                                |  |
|--------------------------------|--|
| SerialPortSelector_SerialPort0 |  |
| NUM_SERIALPORTSELECTOR         |  |

### 6.1.1.152 \_spinSerialPortSourceEnums

enum `_spinSerialPortSourceEnums`

< Specifies the physical input Line on which to receive serial data.

#### Enumerator

|                        |  |
|------------------------|--|
| SerialPortSource_Line0 |  |
| SerialPortSource_Line1 |  |
| SerialPortSource_Line2 |  |
| SerialPortSource_Line3 |  |
| SerialPortSource_Off   |  |
| NUM_SERIALPORTSOURCE   |  |

### 6.1.1.153 \_spinSerialPortStopBitsEnums

enum `_spinSerialPortStopBitsEnums`

< This feature controls the number of stop bits used by the selected serial port.

#### Enumerator

|                                  |  |
|----------------------------------|--|
| SerialPortStopBits_Bits1         |  |
| SerialPortStopBits_Bits1AndAHalf |  |
| SerialPortStopBits_Bits2         |  |
| NUM_SERIALPORTSTOPBITS           |  |

### 6.1.1.154 \_spinSoftwareSignalSelectorEnums

enum `_spinSoftwareSignalSelectorEnums`

< Selects which Software Signal features to control.

#### Enumerator

|                                        |                                                   |
|----------------------------------------|---------------------------------------------------|
| SoftwareSignalSelector_SoftwareSignal0 | Selects the software generated signal to control. |
| SoftwareSignalSelector_SoftwareSignal1 | Selects the software generated signal to control. |
| SoftwareSignalSelector_SoftwareSignal2 | Selects the software generated signal to control. |
| NUM_SOFTWARESIGNALSELECTOR             |                                                   |

**6.1.1.155 \_spinSourceSelectorEnums**

```
enum _spinSourceSelectorEnums
```

< Selects the source to control.

**Enumerator**

|                        |                               |
|------------------------|-------------------------------|
| SourceSelector_Source0 | Selects the data source 0.    |
| SourceSelector_Source1 | Selects the data source 1.    |
| SourceSelector_Source2 | Selects the data source 2.    |
| SourceSelector_All     | Selects all the data sources. |
| NUM_SOURCESELECTOR     |                               |

**6.1.1.156 \_spinTestPatternEnums**

```
enum _spinTestPatternEnums
```

< Selects the type of test pattern that is generated by the device as image source.

**Enumerator**

|                               |                                                                                               |
|-------------------------------|-----------------------------------------------------------------------------------------------|
| TestPattern_Off               | Test pattern is disabled.                                                                     |
| TestPattern_Increment         | Pixel value increments by 1 for each pixel.                                                   |
| TestPattern_SensorTestPattern | A test pattern generated by the image sensor. The pattern varies for different sensor models. |
| NUM_TESTPATTERN               |                                                                                               |

**6.1.1.157 \_spinTestPatternGeneratorSelectorEnums**

```
enum _spinTestPatternGeneratorSelectorEnums
```

< Selects which test pattern generator is controlled by the TestPattern feature.

**Enumerator**

|                                            |                                                                                            |
|--------------------------------------------|--------------------------------------------------------------------------------------------|
| TestPatternGeneratorSelector_Sensor        | TestPattern feature controls the sensor's test pattern generator.                          |
| TestPatternGeneratorSelector_PipelineStart | TestPattern feature controls the test pattern inserted at the start of the image pipeline. |
| NUM_TESTPATTERNGENERATORSELECTOR           |                                                                                            |

**6.1.1.158 \_spinTimerSelectorEnums**

```
enum _spinTimerSelectorEnums
```

< Selects which Timer to configure.

**Enumerator**

|                      |                      |
|----------------------|----------------------|
| TimerSelector_Timer0 | Selects the Timer 0. |
| TimerSelector_Timer1 | Selects the Timer 1. |
| TimerSelector_Timer2 | Selects the Timer 2. |
| NUM_TIMERSELECTOR    |                      |

**6.1.1.159 \_spinTimerStatusEnums**

```
enum _spinTimerStatusEnums
```

< Returns the current status of the Timer.

**Enumerator**

|                              |                                                   |
|------------------------------|---------------------------------------------------|
| TimerStatus_TimerIdle        | The Timer is idle.                                |
| TimerStatus_TimerTriggerWait | The Timer is waiting for a start trigger.         |
| TimerStatus_TimerActive      | The Timer is counting for the specified duration. |
| TimerStatus_TimerCompleted   | The Timer reached the TimerDuration count.        |
| NUM_TIMERSTATUS              |                                                   |

**6.1.1.160 \_spinTimerTriggerActivationEnums**

```
enum _spinTimerTriggerActivationEnums
```

< Selects the activation mode of the trigger to start the Timer.

**Enumerator**

|                                    |                                                                               |
|------------------------------------|-------------------------------------------------------------------------------|
| TimerTriggerActivation_RisingEdge  | Starts counting on the Rising Edge of the selected trigger signal.            |
| TimerTriggerActivation_FallingEdge | Starts counting on the Falling Edge of the selected trigger signal.           |
| TimerTriggerActivation_AnyEdge     | Starts counting on the Falling or Rising Edge of the selected trigger signal. |
| TimerTriggerActivation_LevelHigh   | Counts as long as the selected trigger signal level is High.                  |
| TimerTriggerActivation_LevelLow    | Counts as long as the selected trigger signal level is Low.                   |
| NUM_TIMERTRIGGERACTIVATION         |                                                                               |

### 6.1.1.161 `_spinTimerTriggerSourceEnums`

```
enum _spinTimerTriggerSourceEnums
```

< Selects the source of the trigger to start the Timer.

#### Enumerator

|                                                    |                                                                                                                     |
|----------------------------------------------------|---------------------------------------------------------------------------------------------------------------------|
| <code>TimerTriggerSource_Off</code>                | Disables the Timer trigger.                                                                                         |
| <code>TimerTriggerSource_AcquisitionTrigger</code> | Starts with the reception of the Acquisition Trigger.                                                               |
| <code>TimerTriggerSource_AcquisitionStart</code>   | Starts with the reception of the Acquisition Start.                                                                 |
| <code>TimerTriggerSource_AcquisitionEnd</code>     | Starts with the reception of the Acquisition End.                                                                   |
| <code>TimerTriggerSource_FrameTrigger</code>       | Starts with the reception of the Frame Start Trigger.                                                               |
| <code>TimerTriggerSource_FrameStart</code>         | Starts with the reception of the Frame Start.                                                                       |
| <code>TimerTriggerSource_FrameEnd</code>           | Starts with the reception of the Frame End.                                                                         |
| <code>TimerTriggerSource_FrameBurstStart</code>    | Starts with the reception of the Frame Burst Start.                                                                 |
| <code>TimerTriggerSource_FrameBurstEnd</code>      | Starts with the reception of the Frame Burst End.                                                                   |
| <code>TimerTriggerSource_LineTrigger</code>        | Starts with the reception of the Line Start Trigger.                                                                |
| <code>TimerTriggerSource_LineStart</code>          | Starts with the reception of the Line Start.                                                                        |
| <code>TimerTriggerSource_LineEnd</code>            | Starts with the reception of the Line End.                                                                          |
| <code>TimerTriggerSource_ExposureStart</code>      | Starts with the reception of the Exposure Start.                                                                    |
| <code>TimerTriggerSource_ExposureEnd</code>        | Starts with the reception of the Exposure End.                                                                      |
| <code>TimerTriggerSource_Line0</code>              | Starts when the specified <code>TimerTriggerActivation</code> condition is met on the chosen I/O Line.              |
| <code>TimerTriggerSource_Line1</code>              | Starts when the specified <code>TimerTriggerActivation</code> condition is met on the chosen I/O Line.              |
| <code>TimerTriggerSource_Line2</code>              | Starts when the specified <code>TimerTriggerActivation</code> condition is met on the chosen I/O Line.              |
| <code>TimerTriggerSource_UserOutput0</code>        | Specifies which User Output bit signal to use as internal source for the trigger.                                   |
| <code>TimerTriggerSource_UserOutput1</code>        | Specifies which User Output bit signal to use as internal source for the trigger.                                   |
| <code>TimerTriggerSource_UserOutput2</code>        | Specifies which User Output bit signal to use as internal source for the trigger.                                   |
| <code>TimerTriggerSource_Counter0Start</code>      | Starts with the reception of the Counter Start.                                                                     |
| <code>TimerTriggerSource_Counter1Start</code>      | Starts with the reception of the Counter Start.                                                                     |
| <code>TimerTriggerSource_Counter2Start</code>      | Starts with the reception of the Counter Start.                                                                     |
| <code>TimerTriggerSource_Counter0End</code>        | Starts with the reception of the Counter End.                                                                       |
| <code>TimerTriggerSource_Counter1End</code>        | Starts with the reception of the Counter End.                                                                       |
| <code>TimerTriggerSource_Counter2End</code>        | Starts with the reception of the Counter End.                                                                       |
| <code>TimerTriggerSource_Timer0Start</code>        | Starts with the reception of the Timer Start.                                                                       |
| <code>TimerTriggerSource_Timer1Start</code>        | Starts with the reception of the Timer Start.                                                                       |
| <code>TimerTriggerSource_Timer2Start</code>        | Starts with the reception of the Timer Start.                                                                       |
| <code>TimerTriggerSource_Timer0End</code>          | Starts with the reception of the Timer End. Note that a timer can retrigger itself to achieve a free running Timer. |
| <code>TimerTriggerSource_Timer1End</code>          | Starts with the reception of the Timer End. Note that a timer can retrigger itself to achieve a free running Timer. |
| <code>TimerTriggerSource_Timer2End</code>          | Starts with the reception of the Timer End. Note that a timer can retrigger itself to achieve a free running Timer. |

## Enumerator

|                                    |                                                         |
|------------------------------------|---------------------------------------------------------|
| TimerTriggerSource_Encoder0        | Starts with the reception of the Encoder output signal. |
| TimerTriggerSource_Encoder1        | Starts with the reception of the Encoder output signal. |
| TimerTriggerSource_Encoder2        | Starts with the reception of the Encoder output signal. |
| TimerTriggerSource_SoftwareSignal0 | Starts on the reception of the Software Signal.         |
| TimerTriggerSource_SoftwareSignal1 | Starts on the reception of the Software Signal.         |
| TimerTriggerSource_SoftwareSignal2 | Starts on the reception of the Software Signal.         |
| TimerTriggerSource_Action0         | Starts with the assertion of the chosen action signal.  |
| TimerTriggerSource_Action1         | Starts with the assertion of the chosen action signal.  |
| TimerTriggerSource_Action2         | Starts with the assertion of the chosen action signal.  |
| TimerTriggerSource_LinkTrigger0    | Starts with the reception of the chosen Link Trigger.   |
| TimerTriggerSource_LinkTrigger1    | Starts with the reception of the chosen Link Trigger.   |
| TimerTriggerSource_LinkTrigger2    | Starts with the reception of the chosen Link Trigger.   |
| NUM_TIMERTRIGGERSOURCE             |                                                         |

## 6.1.1.162 \_spinTransferComponentSelectorEnums

```
enum _spinTransferComponentSelectorEnums
```

< Selects the color component for the control of the TransferStreamChannel feature.

## Enumerator

|                                 |                                                                                                                                                                                        |
|---------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| TransferComponentSelector_Red   | The TransferStreamChannel feature controls the index of the stream channel for the streaming of the red plane of the planar pixel formats.                                             |
| TransferComponentSelector_Green | The TransferStreamChannel feature controls the index of the stream channel for the streaming of the green plane of the planar pixel formats.                                           |
| TransferComponentSelector_Blue  | The TransferStreamChannel feature controls the index of the stream channel for the streaming of blue plane of the planar pixel formats.                                                |
| TransferComponentSelector_All   | The TransferStreamChannel feature controls the index of the stream channel for the streaming of all the planes of the planar pixel formats simultaneously or non planar pixel formats. |
| NUM_TRANSFERCOMPONENTSELECTOR   |                                                                                                                                                                                        |

## 6.1.1.163 \_spinTransferControlModeEnums

```
enum _spinTransferControlModeEnums
```

< Selects the control method for the transfers. Basic and Automatic start transmitting data as soon as there is enough data to fill a link layer packet. User Controlled allows you to directly control the transfer of blocks.

## Enumerator

|                                    |                 |
|------------------------------------|-----------------|
| TransferControlMode_Basic          | Basic           |
| TransferControlMode_Automatic      | Automatic       |
| TransferControlMode_UserControlled | User Controlled |
| NUM_TRANSFERCONTROLMODE            |                 |

**6.1.1.164 \_spinTransferOperationModeEnums**

```
enum _spinTransferOperationModeEnums
```

< Selects the operation mode of the transfer. Continuous is similar to Basic/Automatic but you can start/stop the transfer while acquisition runs independently. Multi Block transmits a specified number of blocks and then stops.

## Enumerator

|                                  |             |
|----------------------------------|-------------|
| TransferOperationMode_Continuous | Continuous  |
| TransferOperationMode_MultiBlock | Multi Block |
| NUM_TRANSFEROPERATIONMODE        |             |

**6.1.1.165 \_spinTransferQueueModeEnums**

```
enum _spinTransferQueueModeEnums
```

< Specifies the operation mode of the transfer queue.

## Enumerator

|                                   |                                            |
|-----------------------------------|--------------------------------------------|
| TransferQueueMode_FirstInFirstOut | Blocks first In are transferred Out first. |
| NUM_TRANSFERQUEUEMODE             |                                            |

**6.1.1.166 \_spinTransferSelectorEnums**

```
enum _spinTransferSelectorEnums
```

< Selects which stream transfers are currently controlled by the selected Transfer features.

## Enumerator

|                          |                                                                     |
|--------------------------|---------------------------------------------------------------------|
| TransferSelector_Stream0 | The transfer features control the data stream 0.                    |
| TransferSelector_Stream1 | The transfer features control the data stream 1.                    |
| TransferSelector_Stream2 | The transfer features control the data stream 2.                    |
| TransferSelector_All     | The transfer features control all the data streams simulateneously. |
| NUM_TRANSFERSELECTOR     |                                                                     |



**6.1.1.167 \_spinTransferStatusSelectorEnums**

```
enum _spinTransferStatusSelectorEnums
```

< Selects which status of the transfer module to read.

**Enumerator**

|                                      |                                                                                                                          |
|--------------------------------------|--------------------------------------------------------------------------------------------------------------------------|
| TransferStatusSelector_Streaming     | Data blocks are transmitted when enough data is available.                                                               |
| TransferStatusSelector_Paused        | Data blocks transmission is suspended immediately.                                                                       |
| TransferStatusSelector_Stopping      | Data blocks transmission is stopping. The current block transmission will be completed and the transfer state will stop. |
| TransferStatusSelector_Stopped       | Data blocks transmission is stopped.                                                                                     |
| TransferStatusSelector_QueueOverflow | Data blocks queue is in overflow state.                                                                                  |
| NUM_TRANSFERSTATUSSELECTOR           |                                                                                                                          |

**6.1.1.168 \_spinTransferTriggerActivationEnums**

```
enum _spinTransferTriggerActivationEnums
```

< Specifies the activation mode of the transfer control trigger.

**Enumerator**

|                                       |                                                                                                                                                               |
|---------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| TransferTriggerActivation_RisingEdge  | Specifies that the trigger is considered valid on the rising edge of the source signal.                                                                       |
| TransferTriggerActivation_FallingEdge | Specifies that the trigger is considered valid on the falling edge of the source signal.                                                                      |
| TransferTriggerActivation_AnyEdge     | Specifies that the trigger is considered valid on the falling or rising edge of the source signal.                                                            |
| TransferTriggerActivation_LevelHigh   | Specifies that the trigger is considered valid as long as the level of the source signal is high. This can apply to TransferActive and TransferPause trigger. |
| TransferTriggerActivation_LevelLow    | Specifies that the trigger is considered valid as long as the level of the source signal is low. This can apply to TransferActive and TransferPause trigger.  |
| NUM_TRANSFERTRIGGERACTIVATION         |                                                                                                                                                               |

**6.1.1.169 \_spinTransferTriggerModeEnums**

```
enum _spinTransferTriggerModeEnums
```

< Controls if the selected trigger is active.

## Enumerator

|                         |                                |
|-------------------------|--------------------------------|
| TransferTriggerMode_Off | Disables the selected trigger. |
| TransferTriggerMode_On  | Enable the selected trigger.   |
| NUM_TRANSFERTRIGGERMODE |                                |

6.1.1.170 `_spinTransferTriggerSelectorEnums`

```
enum _spinTransferTriggerSelectorEnums
```

< Selects the type of transfer trigger to configure.

## Enumerator

|                                            |                                                                                                                             |
|--------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------|
| TransferTriggerSelector_TransferStart      | Selects a trigger to start the transfers.                                                                                   |
| TransferTriggerSelector_TransferStop       | Selects a trigger to stop the transfers.                                                                                    |
| TransferTriggerSelector_TransferAbort      | Selects a trigger to abort the transfers.                                                                                   |
| TransferTriggerSelector_TransferPause      | Selects a trigger to pause the transfers.                                                                                   |
| TransferTriggerSelector_TransferResume     | Selects a trigger to Resume the transfers.                                                                                  |
| TransferTriggerSelector_TransferActive     | Selects a trigger to Activate the transfers. This trigger type is used when TriggerActivation is set LevelHigh or levelLow. |
| TransferTriggerSelector_TransferBurstStart | Selects a trigger to start the transfer of a burst of frames specified by TransferBurstCount.                               |
| TransferTriggerSelector_TransferBurstStop  | Selects a trigger to end the transfer of a burst of frames.                                                                 |
| NUM_TRANSFERTRIGGERSELECTOR                |                                                                                                                             |

6.1.1.171 `_spinTransferTriggerSourceEnums`

```
enum _spinTransferTriggerSourceEnums
```

< Specifies the signal to use as the trigger source for transfers.

## Enumerator

|                                     |                                                                                                                                            |
|-------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| TransferTriggerSource_Line0         | Specifies which physical line (or pin) and associated I/O control block to use as external source for the transfer control trigger signal. |
| TransferTriggerSource_Line1         | Specifies which physical line (or pin) and associated I/O control block to use as external source for the transfer control trigger signal. |
| TransferTriggerSource_Line2         | Specifies which physical line (or pin) and associated I/O control block to use as external source for the transfer control trigger signal. |
| TransferTriggerSource_Counter0Start | Specifies which of the Counter signal to use as internal source for the transfer control trigger signal.                                   |
| TransferTriggerSource_Counter1Start | Specifies which of the Counter signal to use as internal source for the transfer control trigger signal.                                   |

## Enumerator

|                                       |                                                                                                          |
|---------------------------------------|----------------------------------------------------------------------------------------------------------|
| TransferTriggerSource_Counter2Start   | Specifies which of the Counter signal to use as internal source for the transfer control trigger signal. |
| TransferTriggerSource_Counter0End     | Specifies which of the Counter signal to use as internal source for the transfer control trigger signal. |
| TransferTriggerSource_Counter1End     | Specifies which of the Counter signal to use as internal source for the transfer control trigger signal. |
| TransferTriggerSource_Counter2End     | Specifies which of the Counter signal to use as internal source for the transfer control trigger signal. |
| TransferTriggerSource_Timer0Start     | Specifies which Timer signal to use as internal source for the transfer control trigger signal.          |
| TransferTriggerSource_Timer1Start     | Specifies which Timer signal to use as internal source for the transfer control trigger signal.          |
| TransferTriggerSource_Timer2Start     | Specifies which Timer signal to use as internal source for the transfer control trigger signal.          |
| TransferTriggerSource_Timer0End       | Specifies which Timer signal to use as internal source for the transfer control trigger signal.          |
| TransferTriggerSource_Timer1End       | Specifies which Timer signal to use as internal source for the transfer control trigger signal.          |
| TransferTriggerSource_Timer2End       | Specifies which Timer signal to use as internal source for the transfer control trigger signal.          |
| TransferTriggerSource_SoftwareSignal0 | Specifies which Software Signal to use as internal source for the transfer control trigger signal.       |
| TransferTriggerSource_SoftwareSignal1 | Specifies which Software Signal to use as internal source for the transfer control trigger signal.       |
| TransferTriggerSource_SoftwareSignal2 | Specifies which Software Signal to use as internal source for the transfer control trigger signal.       |
| TransferTriggerSource_Action0         | Specifies which Action command to use as internal source for the transfer control trigger signal.        |
| TransferTriggerSource_Action1         | Specifies which Action command to use as internal source for the transfer control trigger signal.        |
| TransferTriggerSource_Action2         | Specifies which Action command to use as internal source for the transfer control trigger signal.        |
| NUM_TRANSFERTRIGGERSOURCE             |                                                                                                          |

## 6.1.1.172 \_spinTriggerActivationEnums

```
enum _spinTriggerActivationEnums
```

< Specifies the activation mode of the trigger.

## Enumerator

|                               |  |
|-------------------------------|--|
| TriggerActivation_LevelLow    |  |
| TriggerActivation_LevelHigh   |  |
| TriggerActivation_FallingEdge |  |
| TriggerActivation_RisingEdge  |  |
| TriggerActivation_AnyEdge     |  |
| NUM_TRIGGERACTIVATION         |  |

#### 6.1.1.173 \_spinTriggerModeEnums

enum `_spinTriggerModeEnums`

< Controls whether or not trigger is active.

##### Enumerator

|                 |  |
|-----------------|--|
| TriggerMode_Off |  |
| TriggerMode_On  |  |
| NUM_TRIGGERMODE |  |

#### 6.1.1.174 \_spinTriggerOverlapEnums

enum `_spinTriggerOverlapEnums`

< Specifies the overlap mode of the trigger.

##### Enumerator

|                              |  |
|------------------------------|--|
| TriggerOverlap_Off           |  |
| TriggerOverlap_ReadOut       |  |
| TriggerOverlap_PreviousFrame |  |
| NUM_TRIGGEROVERLAP           |  |

#### 6.1.1.175 \_spinTriggerSelectorEnums

enum `_spinTriggerSelectorEnums`

< Selects the type of trigger to configure.

##### Enumerator

|                                  |  |
|----------------------------------|--|
| TriggerSelector_AcquisitionStart |  |
| TriggerSelector_FrameStart       |  |
| TriggerSelector_FrameBurstStart  |  |
| NUM_TRIGGERSELECTOR              |  |

**6.1.1.176 \_spinTriggerSourceEnums**

```
enum _spinTriggerSourceEnums
```

< Specifies the internal signal or physical input line to use as the trigger source.

**Enumerator**

|                             |  |
|-----------------------------|--|
| TriggerSource_Software      |  |
| TriggerSource_Line0         |  |
| TriggerSource_Line1         |  |
| TriggerSource_Line2         |  |
| TriggerSource_Line3         |  |
| TriggerSource_UserOutput0   |  |
| TriggerSource_UserOutput1   |  |
| TriggerSource_UserOutput2   |  |
| TriggerSource_UserOutput3   |  |
| TriggerSource_Counter0Start |  |
| TriggerSource_Counter1Start |  |
| TriggerSource_Counter0End   |  |
| TriggerSource_Counter1End   |  |
| TriggerSource_LogicBlock0   |  |
| TriggerSource_LogicBlock1   |  |
| TriggerSource_Action0       |  |
| NUM_TRIGGERSOURCE           |  |

**6.1.1.177 \_spinUserOutputSelectorEnums**

```
enum _spinUserOutputSelectorEnums
```

< Selects which bit of the User Output register is set by UserOutputValue.

**Enumerator**

|                                |  |
|--------------------------------|--|
| UserOutputSelector_UserOutput0 |  |
| UserOutputSelector_UserOutput1 |  |
| UserOutputSelector_UserOutput2 |  |
| UserOutputSelector_UserOutput3 |  |
| NUM_USEROUTPUTSELECTOR         |  |

**6.1.1.178 \_spinUserSetDefaultEnums**

```
enum _spinUserSetDefaultEnums
```

< Selects the feature User Set to load and make active by default when the device is restarted.

**Enumerator**

|                         |                          |
|-------------------------|--------------------------|
| UserSetDefault_Default  | Factory default set.     |
| UserSetDefault_UserSet0 | User configurable set 0. |
| UserSetDefault_UserSet1 | User configurable set 1. |
| NUM_USERSETDEFAULT      |                          |

**6.1.1.179 \_spinUserSetSelectorEnums**

```
enum _spinUserSetSelectorEnums
```

< Selects the feature User Set to load, save or configure.

**Enumerator**

|                          |                          |
|--------------------------|--------------------------|
| UserSetSelector_Default  | Factory default set.     |
| UserSetSelector_UserSet0 | User configurable set 0. |
| UserSetSelector_UserSet1 | User configurable set 1. |
| NUM_USERSETSELECTOR      |                          |

**6.1.1.180 \_spinWhiteClipSelectorEnums**

```
enum _spinWhiteClipSelectorEnums
```

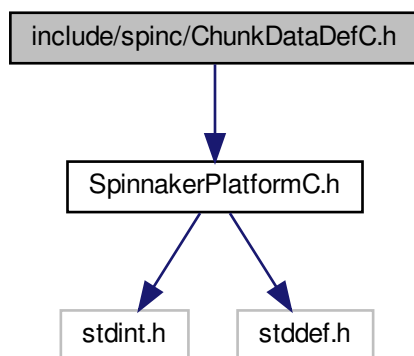
< Selects which White Clip to control.

**Enumerator**

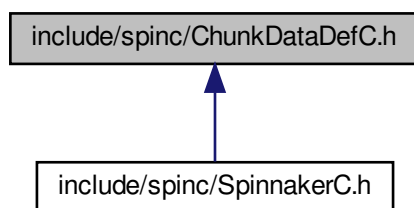
|                         |                                                     |
|-------------------------|-----------------------------------------------------|
| WhiteClipSelector_All   | White Clip will be applied to all channels or taps. |
| WhiteClipSelector_Red   | White Clip will be applied to the red channel.      |
| WhiteClipSelector_Green | White Clip will be applied to the green channel.    |
| WhiteClipSelector_Blue  | White Clip will be applied to the blue channel.     |
| WhiteClipSelector_Y     | White Clip will be applied to Y channel.            |
| WhiteClipSelector_U     | White Clip will be applied to U channel.            |
| WhiteClipSelector_V     | White Clip will be applied to V channel.            |
| WhiteClipSelector_Tap1  | White Clip will be applied to Tap 1.                |
| WhiteClipSelector_Tap2  | White Clip will be applied to Tap 2.                |
| NUM_WHITECLIPSELECTOR   |                                                     |

## 6.2 include/spinc/ChunkDataDefC.h File Reference

Include dependency graph for ChunkDataDefC.h:



This graph shows which files directly or indirectly include this file:



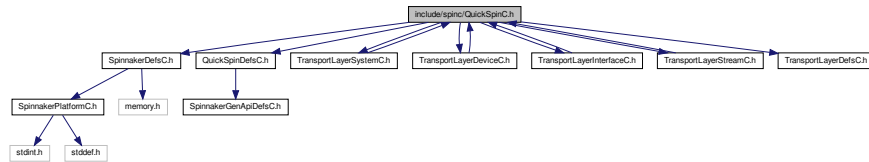
## Data Structures

- [struct `\_spinChunkData`](#)

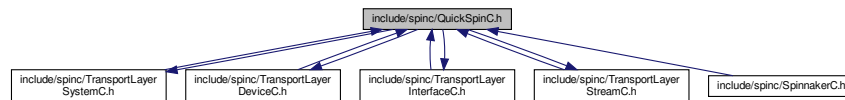
*The type of information that can be obtained from image chunk data.*

## 6.3 include/spinc/QuickSpinC.h File Reference

Include dependency graph for QuickSpinC.h:



This graph shows which files directly or indirectly include this file:



## Functions

- [SPINNAKERC\\_API quickSpinInit](#) ([spinCamera](#) hCamera, quickSpin \*pQuickSpin)
- [SPINNAKERC\\_API quickSpinInitEx](#) ([spinCamera](#) hCamera, quickSpin \*pQuickSpin, quickSpinTLDevice \*pQuickSpinTLDevice, quickSpinTLStream \*pQuickSpinTLStream)
- [SPINNAKERC\\_API quickSpinTLDeviceInit](#) ([spinCamera](#) hCamera, quickSpinTLDevice \*pQuickSpinTLDevice)
- [SPINNAKERC\\_API quickSpinTLStreamInit](#) ([spinCamera](#) hCamera, quickSpinTLStream \*pQuickSpinTLStream)
- [SPINNAKERC\\_API quickSpinTLInterfaceInit](#) ([spinInterface](#) hInterface, quickSpinTLInterface \*pQuickSpinTLInterface)
- [SPINNAKERC\\_API quickSpinTLSystemInit](#) ([spinSystem](#) hSystem, quickSpinTLSystem \*pQuickSpinTLSystem)

### 6.3.1 Function Documentation

#### 6.3.1.1 quickSpinInit()

```

SPINNAKERC_API quickSpinInit (
    spinCamera hCamera,
    quickSpin * pQuickSpin )
  
```



### 6.3.1.2 quickSpinInitEx()

```
SPINNAKERC_API quickSpinInitEx (
    spinCamera hCamera,
    quickSpin * pQuickSpin,
    quickSpinTLDevice * pQuickSpinTLDevice,
    quickSpinTLStream * pQuickSpinTLStream )
```

### 6.3.1.3 quickSpinTLDeviceInit()

```
SPINNAKERC_API quickSpinTLDeviceInit (
    spinCamera hCamera,
    quickSpinTLDevice * pQuickSpinTLDevice )
```

### 6.3.1.4 quickSpinTLInterfaceInit()

```
SPINNAKERC_API quickSpinTLInterfaceInit (
    spinInterface hInterface,
    quickSpinTLInterface * pQuickSpinTLInterface )
```

### 6.3.1.5 quickSpinTLStreamInit()

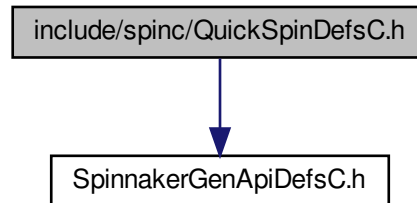
```
SPINNAKERC_API quickSpinTLStreamInit (
    spinCamera hCamera,
    quickSpinTLStream * pQuickSpinTLStream )
```

### 6.3.1.6 quickSpinTLSystemInit()

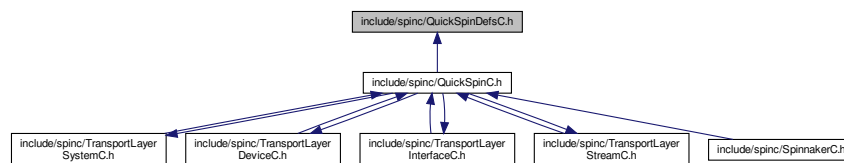
```
SPINNAKERC_API quickSpinTLSystemInit (
    spinSystem hSystem,
    quickSpinTLSystem * pQuickSpinTLSystem )
```

## 6.4 include/spinc/QuickSpinDefsC.h File Reference

Include dependency graph for QuickSpinDefsC.h:



This graph shows which files directly or indirectly include this file:



### Data Structures

- struct [\\_quickSpin](#)

### Typedefs

- typedef [spinNodeHandle](#) [quickSpinStringNode](#)
- typedef [spinNodeHandle](#) [quickSpinIntegerNode](#)
- typedef [spinNodeHandle](#) [quickSpinFloatNode](#)
- typedef [spinNodeHandle](#) [quickSpinBooleanNode](#)
- typedef [spinNodeHandle](#) [quickSpinEnumerationNode](#)
- typedef [spinNodeHandle](#) [quickSpinCommandNode](#)
- typedef [spinNodeHandle](#) [quickSpinRegisterNode](#)

#### 6.4.1 Typedef Documentation

#### 6.4.1.1 quickSpinBooleanNode

```
typedef spinNodeHandle quickSpinBooleanNode
```

#### 6.4.1.2 quickSpinCommandNode

```
typedef spinNodeHandle quickSpinCommandNode
```

#### 6.4.1.3 quickSpinEnumerationNode

```
typedef spinNodeHandle quickSpinEnumerationNode
```

#### 6.4.1.4 quickSpinFloatNode

```
typedef spinNodeHandle quickSpinFloatNode
```

#### 6.4.1.5 quickSpinIntegerNode

```
typedef spinNodeHandle quickSpinIntegerNode
```

#### 6.4.1.6 quickSpinRegisterNode

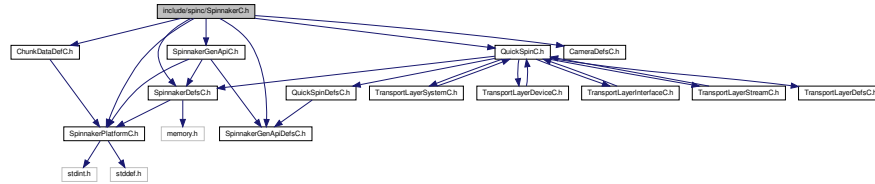
```
typedef spinNodeHandle quickSpinRegisterNode
```

#### 6.4.1.7 quickSpinStringNode

```
typedef spinNodeHandle quickSpinStringNode
```

## 6.5 include/spinc/SpinnakerC.h File Reference

Include dependency graph for SpinnakerC.h:



### Functions

- [SPINNAKERC\\_API spinErrorGetLast](#) (spinError \*pError)  
*Retrieves the error code of the last error.*
- [SPINNAKERC\\_API spinErrorGetLastMessage](#) (char \*pBuf, size\_t \*pBufLen)  
*Retrieves the error message of the last error.*
- [SPINNAKERC\\_API spinErrorGetLastBuildDate](#) (char \*pBuf, size\_t \*pBufLen)  
*Retrieves the build date of the last error.*
- [SPINNAKERC\\_API spinErrorGetLastBuildTime](#) (char \*pBuf, size\_t \*pBufLen)  
*Retrieves the build time of the last error.*
- [SPINNAKERC\\_API spinErrorGetLastFileName](#) (char \*pBuf, size\_t \*pBufLen)  
*Retrieves the filename of the last error.*
- [SPINNAKERC\\_API spinErrorGetLastFullMessage](#) (char \*pBuf, size\_t \*pBufLen)  
*Retrieves the full error message of the last error.*
- [SPINNAKERC\\_API spinErrorGetLastFunctionName](#) (char \*pBuf, size\_t \*pBufLen)  
*Retrieves the function name of the last error.*
- [SPINNAKERC\\_API spinErrorGetLastLineNumber](#) (int64\_t \*pLineNum)  
*Retrieves the line number of the last error.*
- [SPINNAKERC\\_API spinSystemGetInstance](#) (spinSystem \*phSystem)  
*Retrieves an instance of the system object; the system is a singleton, so there will only ever be one instance; system instance must be destroyed by calling spinSystemReleaseInstance.*
- [SPINNAKERC\\_API spinSystemReleaseInstance](#) (spinSystem hSystem)  
*Releases the system; make sure handle is cleaned up properly by setting it to NULL after system is released; the handle can only be used again after calling spinSystemGetInstance.*
- [SPINNAKERC\\_API spinSystemGetInterfaces](#) (spinSystem hSystem, spinInterfaceList hInterfaceList)  
*Retrieves a list of detected (and enumerable) interfaces on the system; interface lists must be created and destroyed.*
- [SPINNAKERC\\_API spinSystemGetCameras](#) (spinSystem hSystem, spinCameraList hCameraList)  
*Retrieves a list of detected (and enumerable) cameras on the system; camera lists must be created and destroyed.*
- [SPINNAKERC\\_API spinSystemGetCamerasEx](#) (spinSystem hSystem, bool8\_t bUpdateInterfaces, bool8\_t bUpdateCameras, spinCameraList hCameraList)  
*Retrieves a list of detected (and enumerable) cameras on the system; manually set whether to update the current interface and camera lists; camera lists must be created and destroyed.*
- [SPINNAKERC\\_API spinSystemSetLoggingLevel](#) (spinSystem hSystem, spinnakerLogLevel logLevel)  
*Sets the logging level for all logging events on the system.*
- [SPINNAKERC\\_API spinSystemGetLoggingLevel](#) (spinSystem hSystem, spinnakerLogLevel \*pLogLevel)  
*Retrieves the logging level for all logging events on the system.*
- [SPINNAKERC\\_API spinSystemRegisterLogEventHandler](#) (spinSystem hSystem, spinLogEventHandler hLogEventHandler)  
*Registers a log event handler for all logging events on the system.*

- Registers a logging event handler to the system (event handlers registered in this way must be unregistered)*

  - [SPINNAKERC\\_API spinSystemUnregisterLogEventHandler](#) ([spinSystem](#) hSystem, [spinLogEventHandler](#) hLogEventHandler)

*Unregisters a selected logging event handler from the system.*
- [SPINNAKERC\\_API spinSystemUnregisterAllLogEventHandlers](#) ([spinSystem](#) hSystem)

*Unregisters all logging event handlers from the system.*
- [SPINNAKERC\\_API spinSystemIsInUse](#) ([spinSystem](#) hSystem, [bool8\\_t](#) \*pbIsInUse)

*Checks whether a system is currently in use.*
- [SPINNAKERC\\_API spinSystemRegisterDeviceArrivalEventHandler](#) ([spinSystem](#) hSystem, [spinDeviceArrivalEventHandler](#) hDeviceArrivalEventHandler)

*Registers a device arrival event handler to every interface on the system (event handlers registered this way must be unregistered)*
- [SPINNAKERC\\_API spinSystemRegisterDeviceRemovalEventHandler](#) ([spinSystem](#) hSystem, [spinDeviceRemovalEventHandler](#) hDeviceRemovalEventHandler)

*Registers a device removal event handler to the system to every interface on the system (event handlers registered this way must be unregistered)*
- [SPINNAKERC\\_API spinSystemUnregisterDeviceArrivalEventHandler](#) ([spinSystem](#) hSystem, [spinDeviceArrivalEventHandler](#) hDeviceArrivalEventHandler)

*Unregisters a device arrival event handler from the system.*
- [SPINNAKERC\\_API spinSystemUnregisterDeviceRemovalEventHandler](#) ([spinSystem](#) hSystem, [spinDeviceRemovalEventHandler](#) hDeviceRemovalEventHandler)

*Unregisters a device removal event handler from the system.*
- [SPINNAKERC\\_API spinSystemRegisterInterfaceEventHandler](#) ([spinSystem](#) hSystem, [spinInterfaceEventHandler](#) hInterfaceEventHandler)

*Registers an interface event handler (device arrival and device removal) to every interface on the system (interface events registered this way must be unregistered) If new interfaces are detected by the system after [spinSystemRegisterInterfaceEventHandler\(\)](#) is called, those interfaces will be automatically registered with this event.*
- [SPINNAKERC\\_API spinSystemUnregisterInterfaceEventHandler](#) ([spinSystem](#) hSystem, [spinInterfaceEventHandler](#) hInterfaceEventHandler)

*Unregisters an interface event handler from the system.*
- [SPINNAKERC\\_API spinSystemUpdateCameras](#) ([spinSystem](#) hSystem, [bool8\\_t](#) \*pbChanged)

*Updates the list of cameras on the system, informing whether there has been any changes.*
- [SPINNAKERC\\_API spinSystemUpdateCamerasEx](#) ([spinSystem](#) hSystem, [bool8\\_t](#) bUpdateInterfaces, [bool8\\_t](#) \*pbChanged)

*Updates the list of cameras on the system, informing whether there has been any changes; manually set whether to update the current interface lists.*
- [SPINNAKERC\\_API spinSystemSendActionCommand](#) ([spinSystem](#) hSystem, [size\\_t](#) iDeviceKey, [size\\_t](#) iGroupKey, [size\\_t](#) iGroupMask, [size\\_t](#) iActionTime, [size\\_t](#) \*piResultSize, [actionCommandResult](#) results[])

*Broadcast an Action Command to all devices on system.*
- [SPINNAKERC\\_API spinSystemGetLibraryVersion](#) ([spinSystem](#) hSystem, [spinLibraryVersion](#) \*hLibraryVersion)

*Get current library version of Spinnaker.*
- [SPINNAKERC\\_API spinSystemGetTLNodeMap](#) ([spinSystem](#) hSystem, [spinNodeMapHandle](#) \*phNodeMap)

*Retrieves the transport layer nodemap from the system.*
- [SPINNAKERC\\_API spinInterfaceListCreateEmpty](#) ([spinInterfaceList](#) \*phInterfaceList)

*Creates an empty interface list (interface lists created this way must be destroyed)*
- [SPINNAKERC\\_API spinInterfaceListDestroy](#) ([spinInterfaceList](#) hInterfaceList)

*Destroys an interface list.*
- [SPINNAKERC\\_API spinInterfaceListGetSize](#) ([spinInterfaceList](#) hInterfaceList, [size\\_t](#) \*pSize)

*Retrieves the number of interfaces in an interface list.*
- [SPINNAKERC\\_API spinInterfaceListGet](#) ([spinInterfaceList](#) hInterfaceList, [size\\_t](#) index, [spinInterface](#) \*phInterface)

*Retrieves an interface from an interface list using an index (interfaces retrieved this way must be released)*

- [SPINNAKERC\\_API spinInterfaceListClear](#) ([spinInterfaceList](#) hInterfaceList)  
*Clears an interface list.*
- [SPINNAKERC\\_API spinCameraListCreateEmpty](#) ([spinCameraList](#) \*phCameraList)  
*Creates an empty camera list (camera lists created this way must be destroyed)*
- [SPINNAKERC\\_API spinCameraListDestroy](#) ([spinCameraList](#) hCameraList)  
*Destroys a camera list.*
- [SPINNAKERC\\_API spinCameraListGetSize](#) ([spinCameraList](#) hCameraList, [size\\_t](#) \*pSize)  
*Retrieves the number of cameras on a camera list.*
- [SPINNAKERC\\_API spinCameraListGet](#) ([spinCameraList](#) hCameraList, [size\\_t](#) index, [spinCamera](#) \*phCamera)  
*Retrieves a camera from a camera list using an index.*
- [SPINNAKERC\\_API spinCameraListClear](#) ([spinCameraList](#) hCameraList)  
*Clears a camera list.*
- [SPINNAKERC\\_API spinCameraListRemove](#) ([spinCameraList](#) hCameraList, [size\\_t](#) index)  
*Removes a camera from a camera list using its index.*
- [SPINNAKERC\\_API spinCameraListAppend](#) ([spinCameraList](#) hCameraListBase, [spinCameraList](#) hCameraListToAppend)  
*Appends all the cameras from one camera list to another.*
- [SPINNAKERC\\_API spinCameraListGetBySerial](#) ([spinCameraList](#) hCameraList, [const char](#) \*pSerial, [spinCamera](#) \*phCamera)  
*Retrieves a camera from a camera list using its serial number.*
- [SPINNAKERC\\_API spinCameraListRemoveBySerial](#) ([spinCameraList](#) hCameraList, [const char](#) \*pSerial)  
*Removes a camera from a camera list using its serial number.*
- [SPINNAKERC\\_API spinInterfaceUpdateCameras](#) ([spinInterface](#) hInterface, [bool8\\_t](#) \*pbChanged)  
*Checks whether any cameras have been connected or disconnected on an interface.*
- [SPINNAKERC\\_API spinInterfaceGetCameras](#) ([spinInterface](#) hInterface, [spinCameraList](#) hCameraList)  
*Retrieves a camera list from an interface; camera lists must be created and destroy.*
- [SPINNAKERC\\_API spinInterfaceGetCamerasEx](#) ([spinInterface](#) hInterface, [bool8\\_t](#) bUpdateCameras, [spinCameraList](#) hCameraList)  
*Retrieves a camera list from an interface; manually set whether to update the cameras; camera lists must be created and destroyed.*
- [SPINNAKERC\\_API spinInterfaceGetTLNodeMap](#) ([spinInterface](#) hInterface, [spinNodeMapHandle](#) \*phNodeMap)  
*Retrieves the transport layer nodemap from an interface.*
- [SPINNAKERC\\_API spinInterfaceRegisterDeviceArrivalEventHandler](#) ([spinInterface](#) hInterface, [spinDeviceArrivalEventHandler](#) hDeviceArrivalEventHandler)  
*Registers a device arrival event handler on an interface (event handlers registered in this way must be unregistered)*
- [SPINNAKERC\\_API spinInterfaceRegisterDeviceRemovalEventHandler](#) ([spinInterface](#) hInterface, [spinDeviceRemovalEventHandler](#) hDeviceRemovalEventHandler)  
*Registers a device removal event handler on an interface (event handlers registered in this way must be unregistered)*
- [SPINNAKERC\\_API spinInterfaceUnregisterDeviceArrivalEventHandler](#) ([spinInterface](#) hInterface, [spinDeviceArrivalEventHandler](#) hDeviceArrivalEventHandler)  
*Unregisters a device arrival event handler from an interface.*
- [SPINNAKERC\\_API spinInterfaceUnregisterDeviceRemovalEventHandler](#) ([spinInterface](#) hInterface, [spinDeviceRemovalEventHandler](#) hDeviceRemovalEventHandler)  
*Unregisters a device removal event handler from an interface.*
- [SPINNAKERC\\_API spinInterfaceRegisterInterfaceEventHandler](#) ([spinInterface](#) hInterface, [spinInterfaceEventHandler](#) hInterfaceEventHandler)  
*Registers an interface event handler (both device arrival and device removal) on an interface.*
- [SPINNAKERC\\_API spinInterfaceUnregisterInterfaceEventHandler](#) ([spinInterface](#) hInterface, [spinInterfaceEventHandler](#) hInterfaceEventHandler)  
*Unregisters an interface event handler from an interface.*

- [SPINNAKERC\\_API spinInterfaceRelease](#) ([spinInterface](#) hInterface)  
*Releases an interface.*
- [SPINNAKERC\\_API spinInterfaceIsInUse](#) ([spinInterface](#) hInterface, [bool8\\_t](#) \*pbIsInUse)  
*Checks whether an interface is in use.*
- [SPINNAKERC\\_API spinInterfaceSendActionCommand](#) ([spinInterface](#) hInterface, [size\\_t](#) iDeviceKey, [size\\_t](#) iGroupKey, [size\\_t](#) iGroupMask, [size\\_t](#) iActionTime, [size\\_t](#) \*piResultSize, [actionCommandResult](#) results[])  
*Broadcast an Action Command to all devices on interface.*
- [SPINNAKERC\\_API spinCameraInit](#) ([spinCamera](#) hCamera)  
*Initializes a camera, allowing for much more interaction.*
- [SPINNAKERC\\_API spinCameraDeInit](#) ([spinCamera](#) hCamera)  
*Deinitializes a camera, greatly reducing functionality.*
- [SPINNAKERC\\_API spinCameraGetNodeMap](#) ([spinCamera](#) hCamera, [spinNodeMapHandle](#) \*phNodeMap)  
*Retrieves the GenICam nodemap from a camera.*
- [SPINNAKERC\\_API spinCameraGetTLDeviceNodeMap](#) ([spinCamera](#) hCamera, [spinNodeMapHandle](#) \*phNodeMap)  
*Retrieves the transport layer device nodemap from a camera.*
- [SPINNAKERC\\_API spinCameraGetTLStreamNodeMap](#) ([spinCamera](#) hCamera, [spinNodeMapHandle](#) \*phNodeMap)  
*Retrieves the transport layer stream nodemap from a camera.*
- [SPINNAKERC\\_API spinCameraGetAccessMode](#) ([spinCamera](#) hCamera, [spinAccessMode](#) \*pAccessMode)  
*Retrieves the access mode of a camera (as an enum, [spinAccessMode](#))*
- [SPINNAKERC\\_API spinCameraReadPort](#) ([spinCamera](#) hCamera, [uint64\\_t](#) iAddress, [void](#) \*pBuffer, [size\\_t](#) iSize)  
*Reads data from a camera port.*
- [SPINNAKERC\\_API spinCameraWritePort](#) ([spinCamera](#) hCamera, [uint64\\_t](#) iAddress, [void](#) \*pBuffer, [size\\_t](#) iSize)  
*Writes data to a camera port.*
- [SPINNAKERC\\_API spinCameraBeginAcquisition](#) ([spinCamera](#) hCamera)  
*Has a camera start acquiring images.*
- [SPINNAKERC\\_API spinCameraEndAcquisition](#) ([spinCamera](#) hCamera)  
*Has a camera stop acquiring images.*
- [SPINNAKERC\\_API spinCameraGetNextImage](#) ([spinCamera](#) hCamera, [spinImage](#) \*phImage)  
*Retrieves an image from a camera.*
- [SPINNAKERC\\_API spinCameraGetNextImageEx](#) ([spinCamera](#) hCamera, [uint64\\_t](#) grabTimeout, [spinImage](#) \*phImage)  
*Retrieves an image from a camera; manually set the timeout in milliseconds.*
- [SPINNAKERC\\_API spinCameraGetUniqueID](#) ([spinCamera](#) hCamera, [char](#) \*pBuf, [size\\_t](#) \*pBufLen)  
*Retrieves a unique identifier for a camera.*
- [SPINNAKERC\\_API spinCameraIsStreaming](#) ([spinCamera](#) hCamera, [bool8\\_t](#) \*pbIsStreaming)  
*Checks whether a camera is currently acquiring images.*
- [SPINNAKERC\\_API spinCameraGetGuiXml](#) ([spinCamera](#) hCamera, [char](#) \*pBuf, [size\\_t](#) \*pBufLen)  
*Retrieves the GUI XML from a camera.*
- [SPINNAKERC\\_API spinCameraRegisterDeviceEventHandler](#) ([spinCamera](#) hCamera, [spinDeviceEventHandler](#) hDeviceEventHandler)  
*Registers a universal device event handler (every device event type) to a camera.*
- [SPINNAKERC\\_API spinCameraRegisterDeviceEventHandlerEx](#) ([spinCamera](#) hCamera, [spinDeviceEventHandler](#) hDeviceEventHandler, [const char](#) \*pName)  
*Registers a specific device event handler (only one device event type) to a camera.*
- [SPINNAKERC\\_API spinCameraUnregisterDeviceEventHandler](#) ([spinCamera](#) hCamera, [spinDeviceEventHandler](#) hDeviceEventHandler)  
*Unregisters a device event handler from a camera.*
- [SPINNAKERC\\_API spinCameraRegisterImageEventHandler](#) ([spinCamera](#) hCamera, [spinImageEventHandler](#) hImageEventHandler)  
*Registers an image event handler to a camera.*

- [SPINNAKERC\\_API spinCameraUnregisterImageEventHandler](#) ([spinCamera](#) hCamera, [spinImageEventHandler](#) hImageEventHandler)  
*Unregisters an image event handler from a camera.*
- [SPINNAKERC\\_API spinCameraRelease](#) ([spinCamera](#) hCamera)  
*Releases a camera.*
- [SPINNAKERC\\_API spinCamerasValid](#) ([spinCamera](#) hCamera, [bool8\\_t](#) \*pbValid)  
*Checks whether a camera is still valid for use.*
- [SPINNAKERC\\_API spinCamerasInitialized](#) ([spinCamera](#) hCamera, [bool8\\_t](#) \*pbInit)  
*Checks whether a camera is currently initialized.*
- [SPINNAKERC\\_API spinCameraDiscoverMaxPacketSize](#) ([spinCamera](#) hCamera, unsigned int \*pMaxPacketSize)  
*Returns the largest packet size that can be safely used on the interface that device is connected to.*
- [SPINNAKERC\\_API spinCameraForceIP](#) ()  
*Forces the camera to be on the same subnet as its corresponding interface.*
- [SPINNAKERC\\_API spinImageCreateEmpty](#) ([spinImage](#) \*phImage)  
*Creates an empty image; images created this way must be destroyed.*
- [SPINNAKERC\\_API spinImageCreate](#) ([spinImage](#) hSrcImage, [spinImage](#) \*phDestImage)  
*Creates an image from another; images created this way must be destroyed.*
- [SPINNAKERC\\_API spinImageCreateEx](#) ([spinImage](#) \*phImage, size\_t width, size\_t height, size\_t offsetX, size\_t offsetY, [spinPixelFormatEnums](#) pixelFormat, void \*pData)  
*Creates an image with some set properties; images created this way must be destroyed.*
- [SPINNAKERC\\_API spinImageDestroy](#) ([spinImage](#) hImage)  
*Destroys an image.*
- [SPINNAKERC\\_API spinImageSetDefaultColorProcessing](#) ([spinColorProcessingAlgorithm](#) algorithm)  
*Sets the default color processing algorithm of all images (if not otherwise set)*
- [SPINNAKERC\\_API spinImageGetDefaultColorProcessing](#) ([spinColorProcessingAlgorithm](#) \*pAlgorithm)  
*Retrieves the default color processing algorithm.*
- [SPINNAKERC\\_API spinImageGetColorProcessing](#) ([spinImage](#) hImage, [spinColorProcessingAlgorithm](#) \*pAlgorithm)  
*Retrieves the color processing algorithm of a specific image.*
- [SPINNAKERC\\_API spinImageConvert](#) ([spinImage](#) hSrcImage, [spinPixelFormatEnums](#) pixelFormat, [spinImage](#) hDestImage)  
*Converts the pixel format of one image into a new image.*
- [SPINNAKERC\\_API spinImageConvertEx](#) ([spinImage](#) hSrcImage, [spinPixelFormatEnums](#) pixelFormat, [spinColorProcessingAlgorithm](#) algorithm, [spinImage](#) hDestImage)  
*Converts the pixel format and color processing algorithm of one image into a new image.*
- [SPINNAKERC\\_API spinImageReset](#) ([spinImage](#) hImage, size\_t width, size\_t height, size\_t offsetX, size\_t offsetY, [spinPixelFormatEnums](#) pixelFormat)  
*Resets an image with some set properties.*
- [SPINNAKERC\\_API spinImageResetEx](#) ([spinImage](#) hImage, size\_t width, size\_t height, size\_t offsetX, size\_t offsetY, [spinPixelFormatEnums](#) pixelFormat, void \*pData)  
*Resets an image with some set properties and image data.*
- [SPINNAKERC\\_API spinImageGetID](#) ([spinImage](#) hImage, [uint64\\_t](#) \*pId)  
*Retrieves the ID of an image.*
- [SPINNAKERC\\_API spinImageGetData](#) ([spinImage](#) hImage, void \*\*ppData)  
*Retrieves the image data of an image.*
- [SPINNAKERC\\_API spinImageGetPrivateData](#) ([spinImage](#) hImage, void \*\*ppData)  
*Retrieves the private data of an image.*
- [SPINNAKERC\\_API spinImageGetBufferSize](#) ([spinImage](#) hImage, size\_t \*pSize)  
*Retrieves the buffer size of an image.*
- [SPINNAKERC\\_API spinImageDeepCopy](#) ([spinImage](#) hSrcImage, [spinImage](#) hDestImage)  
*Creates a deep copy of an image (the destination image must be created as an empty image prior to the deep copy)*



- [SPINNAKERC\\_API spinImageGetWidth](#) ([spinImage](#) hImage, [size\\_t](#) \*pWidth)  
*Retrieves the width of an image.*
- [SPINNAKERC\\_API spinImageGetHeight](#) ([spinImage](#) hImage, [size\\_t](#) \*pHeight)  
*Retrieves the height of an image.*
- [SPINNAKERC\\_API spinImageGetOffsetX](#) ([spinImage](#) hImage, [size\\_t](#) \*pOffsetX)  
*Retrieves the offset of an image along its X axis.*
- [SPINNAKERC\\_API spinImageGetOffsetY](#) ([spinImage](#) hImage, [size\\_t](#) \*pOffsetY)  
*Retrieves the offset of an image along its Y axis.*
- [SPINNAKERC\\_API spinImageGetPaddingX](#) ([spinImage](#) hImage, [size\\_t](#) \*pPaddingX)  
*Retrieves the padding of an image along its X axis.*
- [SPINNAKERC\\_API spinImageGetPaddingY](#) ([spinImage](#) hImage, [size\\_t](#) \*pPaddingY)  
*Retrieves the padding of an image along its Y axis.*
- [SPINNAKERC\\_API spinImageGetFrameID](#) ([spinImage](#) hImage, [uint64\\_t](#) \*pFrameID)  
*Retrieves the frame ID of an image.*
- [SPINNAKERC\\_API spinImageGetTimeStamp](#) ([spinImage](#) hImage, [uint64\\_t](#) \*pTimeStamp)  
*Retrieves the timestamp of an image.*
- [SPINNAKERC\\_API spinImageGetPayloadType](#) ([spinImage](#) hImage, [size\\_t](#) \*pPayloadType)  
*Retrieves the payload type of an image (as an enum, [spinPayloadTypeInfolDs](#))*
- [SPINNAKERC\\_API spinImageGetTLPayloadType](#) ([spinImage](#) hImage, [spinPayloadTypeInfolDs](#) \*pPayloadType)  
*Retrieves the transport layer payload type of an image (as an enum, [spinPayloadTypeInfolDs](#))*
- [SPINNAKERC\\_API spinImageGetPixelFormat](#) ([spinImage](#) hImage, [spinPixelFormatEnums](#) \*pPixelFormat)  
*Retrieves the pixel format of an image (as an enum, [spinPixelFormatEnums](#))*
- [SPINNAKERC\\_API spinImageGetTLPixelFormat](#) ([spinImage](#) hImage, [uint64\\_t](#) \*pPixelFormat)  
*Retrieves the transport layer pixel format of an image (as an unsigned integer)*
- [SPINNAKERC\\_API spinImageGetTLPixelFormatNamespace](#) ([spinImage](#) hImage, [spinPixelFormatNamespaceID](#) \*pPixelFormatNamespace)  
*Retrieves the transport layer pixel format namespace of an image (as an enum, [spinPixelFormatNamespaceID](#))*
- [SPINNAKERC\\_API spinImageGetPixelFormatName](#) ([spinImage](#) hImage, [char](#) \*pBuf, [size\\_t](#) \*pBufLen)  
*Retrieves the pixel format of an image (as a symbolic)*
- [SPINNAKERC\\_API spinImageIsIncomplete](#) ([spinImage](#) hImage, [bool8\\_t](#) \*pIsIncomplete)  
*Checks whether an image is incomplete.*
- [SPINNAKERC\\_API spinImageGetValidPayloadSize](#) ([spinImage](#) hImage, [size\\_t](#) \*pSize)  
*Retrieves the valid payload size of an image.*
- [SPINNAKERC\\_API spinImageSave](#) ([spinImage](#) hImage, [const char](#) \*pFilename, [spinImageFileFormat](#) format)  
*Saves an image using a specified file format (using an enum, [spinImageFileFormat](#))*
- [SPINNAKERC\\_API spinImageSaveFromExt](#) ([spinImage](#) hImage, [const char](#) \*pFilename)  
*Saves an image using a specified file format (using the extension of the filename)*
- [SPINNAKERC\\_API spinImageSavePng](#) ([spinImage](#) hImage, [const char](#) \*pFilename, [const spinPNGOption](#) \*pOption)  
*Saves an image as a PNG image.*
- [SPINNAKERC\\_API spinImageSavePpm](#) ([spinImage](#) hImage, [const char](#) \*pFilename, [const spinPPMOption](#) \*pOption)  
*Saves an image as a PPM image.*
- [SPINNAKERC\\_API spinImageSavePgm](#) ([spinImage](#) hImage, [const char](#) \*pFilename, [const spinPGMOption](#) \*pOption)  
*Saves an image as an PGM image.*
- [SPINNAKERC\\_API spinImageSaveTiff](#) ([spinImage](#) hImage, [const char](#) \*pFilename, [const spinTIFFOption](#) \*pOption)  
*Saves an image as a TIFF image.*

- [SPINNAKERC\\_API spinImageSaveJpeg](#) ([spinImage](#) hImage, const char \*pFilename, const [spinJPEGOption](#) \*pOption)  
*Saves an image as a JPEG image.*
- [SPINNAKERC\\_API spinImageSaveJpg2](#) ([spinImage](#) hImage, const char \*pFilename, const [spinJPG2Option](#) \*pOption)  
*Saves an image as a JPEG 2000 image.*
- [SPINNAKERC\\_API spinImageSaveBmp](#) ([spinImage](#) hImage, const char \*pFilename, const [spinBMPOption](#) \*pOption)  
*Saves an image as a BMP image.*
- [SPINNAKERC\\_API spinImageGetChunkLayoutID](#) ([spinImage](#) hImage, [uint64\\_t](#) \*pId)  
*Retrieves the chunk layout ID of an image.*
- [SPINNAKERC\\_API spinImageCalculateStatistics](#) ([spinImage](#) hImage, const [spinImageStatistics](#) hStatistics)  
*Calculates the image statistics of an image.*
- [SPINNAKERC\\_API spinImageGetStatus](#) ([spinImage](#) hImage, [spinImageStatus](#) \*pStatus)  
*Retrieves the image status of an image.*
- [SPINNAKERC\\_API spinImageGetStatusDescription](#) ([spinImageStatus](#) status, char \*pBuf, [size\\_t](#) \*pBufLen)  
*Retrieves the description of image status.*
- [SPINNAKERC\\_API spinImageRelease](#) ([spinImage](#) hImage)  
*Releases an image.*
- [SPINNAKERC\\_API spinImageHasCRC](#) ([spinImage](#) hImage, [bool8\\_t](#) \*pbHasCRC)  
*Checks whether an image has CRC.*
- [SPINNAKERC\\_API spinImageCheckCRC](#) ([spinImage](#) hImage, [bool8\\_t](#) \*pbCheckCRC)  
*Checks whether the CRC of an image is correct.*
- [SPINNAKERC\\_API spinImageGetBitsPerPixel](#) ([spinImage](#) hImage, [size\\_t](#) \*pBitsPerPixel)  
*Retrieves the number of bits per pixel of an image.*
- [SPINNAKERC\\_API spinImageGetSize](#) ([spinImage](#) hImage, [size\\_t](#) \*pImageSize)  
*Retrieves the size of an image.*
- [SPINNAKERC\\_API spinImageGetStride](#) ([spinImage](#) hImage, [size\\_t](#) \*pStride)  
*Retrieves the stride of an image.*
- [SPINNAKERC\\_API spinDeviceEventHandlerCreate](#) ([spinDeviceEventHandler](#) \*phDeviceEventHandler, [spinDeviceEventFunction](#) pFunction, void \*pUserData)  
*Creates a device event handler.*
- [SPINNAKERC\\_API spinDeviceEventHandlerDestroy](#) ([spinDeviceEventHandler](#) hDeviceEventHandler)  
*Destroys a device event handler.*
- [SPINNAKERC\\_API spinImageEventHandlerCreate](#) ([spinImageEventHandler](#) \*phImageEventHandler, [spinImageEventFunction](#) pFunction, void \*pUserData)  
*Creates an image event handler.*
- [SPINNAKERC\\_API spinImageEventHandlerDestroy](#) ([spinImageEventHandler](#) hImageEventHandler)  
*Destroys an image event handler.*
- [SPINNAKERC\\_API spinDeviceArrivalEventHandlerCreate](#) ([spinDeviceArrivalEventHandler](#) \*phDeviceArrivalEventHandler, [spinArrivalEventFunction](#) pFunction, void \*pUserData)  
*Creates a device arrival event handler.*
- [SPINNAKERC\\_API spinDeviceArrivalEventHandlerDestroy](#) ([spinDeviceArrivalEventHandler](#) hDeviceArrivalEventHandler)  
*Destroys a device arrival event handler.*
- [SPINNAKERC\\_API spinDeviceRemovalEventHandlerCreate](#) ([spinDeviceRemovalEventHandler](#) \*phDeviceRemovalEventHandler, [spinRemovalEventFunction](#) pFunction, void \*pUserData)  
*Creates a device removal event handler.*
- [SPINNAKERC\\_API spinDeviceRemovalEventHandlerDestroy](#) ([spinDeviceRemovalEventHandler](#) hDeviceRemovalEventHandler)  
*Destroys a device removal event handler.*

- [SPINNAKERC\\_API spinInterfaceEventHandlerCreate](#) ([spinInterfaceEventHandler](#) \*phInterfaceEvent↔  
Handler, [spinArrivalEventFunction](#) pArrivalFunction, [spinRemovalEventFunction](#) pRemovalFunction, void  
\*pUserData)  
*Creates an interface event handler (both device arrival and device removal)*
- [SPINNAKERC\\_API spinInterfaceEventHandlerDestroy](#) ([spinInterfaceEventHandler](#) hInterfaceEventHandler)  
*Destroys an interface event handler (both device arrival and device removal)*
- [SPINNAKERC\\_API spinLogEventHandlerCreate](#) ([spinLogEventHandler](#) \*phLogEventHandler, [spinLogEventFunction](#)  
pFunction, void \*pUserData)  
*Creates a log event handler.*
- [SPINNAKERC\\_API spinLogEventHandlerDestroy](#) ([spinLogEventHandler](#) hLogEventHandler)  
*Destroys a log event handler.*
- [SPINNAKERC\\_API spinImageStatisticsCreate](#) ([spinImageStatistics](#) \*phStatistics)  
*Creates an image statistics context.*
- [SPINNAKERC\\_API spinImageStatisticsDestroy](#) ([spinImageStatistics](#) hStatistics)  
*Destroys an image statistics context.*
- [SPINNAKERC\\_API spinImageStatisticsEnableAll](#) ([spinImageStatistics](#) hStatistics)  
*Enables all channels of an image statistics context.*
- [SPINNAKERC\\_API spinImageStatisticsDisableAll](#) ([spinImageStatistics](#) hStatistics)  
*Disables all channels of an image statistics context.*
- [SPINNAKERC\\_API spinImageStatisticsEnableGreyOnly](#) ([spinImageStatistics](#) hStatistics)  
*Disables all channels of an image statistics context except grey-scale.*
- [SPINNAKERC\\_API spinImageStatisticsEnableRgbOnly](#) ([spinImageStatistics](#) hStatistics)  
*Disables all channels of an image statistics context except red, blue, and green.*
- [SPINNAKERC\\_API spinImageStatisticsEnableHslOnly](#) ([spinImageStatistics](#) hStatistics)  
*Disables all channels of an image statistics context except hue, saturation, and lightness.*
- [SPINNAKERC\\_API spinImageStatisticsGetChannelStatus](#) ([spinImageStatistics](#) hStatistics, [spinStatistics](#)↔  
Channel channel, [bool8\\_t](#) \*pbEnabled)  
*Checks whether an image statistics context is enabled.*
- [SPINNAKERC\\_API spinImageStatisticsSetChannelStatus](#) ([spinImageStatistics](#) hStatistics, [spinStatistics](#)↔  
Channel channel, [bool8\\_t](#) bEnable)  
*Sets the status of an image statistics channel.*
- [SPINNAKERC\\_API spinImageStatisticsGetRange](#) ([spinImageStatistics](#) hStatistics, [spinStatisticsChannel](#)  
channel, unsigned int \*pMin, unsigned int \*pMax)  
*Retrieves the range of an image statistics channel.*
- [SPINNAKERC\\_API spinImageStatisticsGetPixelValueRange](#) ([spinImageStatistics](#) hStatistics, [spin](#)↔  
[StatisticsChannel](#) channel, unsigned int \*pMin, unsigned int \*pMax)  
*Retrieves the pixel value range of an image statistics channel.*
- [SPINNAKERC\\_API spinImageStatisticsGetNumPixelValues](#) ([spinImageStatistics](#) hStatistics, [spinStatistics](#)↔  
Channel channel, unsigned int \*pNumValues)  
*Retrieves the number of pixel values of an image statistics channel.*
- [SPINNAKERC\\_API spinImageStatisticsGetMean](#) ([spinImageStatistics](#) hStatistics, [spinStatisticsChannel](#)  
channel, float \*pMean)  
*Retrieves the mean of pixel values of an image statistics channel.*
- [SPINNAKERC\\_API spinImageStatisticsGetHistogram](#) ([spinImageStatistics](#) hStatistics, [spinStatisticsChannel](#)  
channel, int \*\*ppHistogram)  
*Retrieves a histogram of an image statistics channel.*
- [SPINNAKERC\\_API spinImageStatisticsGetAll](#) ([spinImageStatistics](#) hStatistics, [spinStatisticsChannel](#) chan-  
nel, unsigned int \*pRangeMin, unsigned int \*pRangeMax, unsigned int \*pPixelValueMin, unsigned int \*p↔  
PixelValueMax, unsigned int \*pNumPixelValues, float \*pPixelValueMean, int \*\*ppHistogram)  
*Retrieves all available information of an image statistics channel.*
- [SPINNAKERC\\_API spinLogDataGetCategoryName](#) ([spinLogEventData](#) hLogEventData, char \*pBuf, [size\\_t](#)  
\*pBufLen)

- Retrieves the category name of a log event.*
- [SPINNAKERC\\_API spinLogDataGetPriority](#) ([spinLogEventData](#) hLogEventData, int64\_t \*pValue)
- Retrieves the priority of a log event.*
- [SPINNAKERC\\_API spinLogDataGetPriorityName](#) ([spinLogEventData](#) hLogEventData, char \*pBuf, size\_t \*pBufLen)
- Retrieves the priority name of a log event.*
- [SPINNAKERC\\_API spinLogDataGetTimestamp](#) ([spinLogEventData](#) hLogEventData, char \*pBuf, size\_t \*pBufLen)
- Retrieves the timestamp of a log event.*
- [SPINNAKERC\\_API spinLogDataGetNDC](#) ([spinLogEventData](#) hLogEventData, char \*pBuf, size\_t \*pBufLen)
- Retrieves the NDC of a log event.*
- [SPINNAKERC\\_API spinLogDataGetThreadName](#) ([spinLogEventData](#) hLogEventData, char \*pBuf, size\_t \*pBufLen)
- Retrieves the thread name of a log event.*
- [SPINNAKERC\\_API spinLogDataGetLogMessage](#) ([spinLogEventData](#) hLogEventData, char \*pBuf, size\_t \*pBufLen)
- Retrieves the log message of a log event.*
- [SPINNAKERC\\_API spinDeviceEventGetId](#) ([spinDeviceEventData](#) hDeviceEventData, uint64\_t \*pEventId)
- Retrieves the event ID of a device event.*
- [SPINNAKERC\\_API spinDeviceEventGetPayloadData](#) ([spinDeviceEventData](#) hDeviceEventData, const uint8\_t \*pBuf, size\_t \*pBufSize)
- Retrieves the payload data of a device event.*
- [SPINNAKERC\\_API spinDeviceEventGetPayloadDataSize](#) ([spinDeviceEventData](#) hDeviceEventData, size\_t \*pBufSize)
- Retrieves the payload data size of a device event.*
- [SPINNAKERC\\_API spinDeviceEventGetName](#) ([spinDeviceEventData](#) hDeviceEventData, char \*pBuf, size\_t \*pBufLen)
- Retrieves the event name of a device event.*
- [SPINNAKERC\\_API spinImageChunkDataGetIntValue](#) ([spinImage](#) hImage, const char \*pName, int64\_t \*pValue)
- [SPINNAKERC\\_API spinImageChunkDataGetFloatValue](#) ([spinImage](#) hImage, const char \*pName, double \*pValue)

## 6.5.1 Function Documentation

### 6.5.1.1 spinCameraBeginAcquisition()

```
SPINNAKERC_API spinCameraBeginAcquisition (
    spinCamera hCamera )
```

Has a camera start acquiring images.

See also

spinError

## Parameters

|                |                                      |
|----------------|--------------------------------------|
| <i>hCamera</i> | The camera to begin acquiring images |
|----------------|--------------------------------------|

## Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

### 6.5.1.2 spinCameraDeInit()

```
SPINNAKERC_API spinCameraDeInit (  
    spinCamera hCamera )
```

Deinitializes a camera, greatly reducing functionality.

## See also

spinError

## Parameters

|                |                            |
|----------------|----------------------------|
| <i>hCamera</i> | The camera to deinitialize |
|----------------|----------------------------|

## Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

### 6.5.1.3 spinCameraDiscoverMaxPacketSize()

```
SPINNAKERC_API spinCameraDiscoverMaxPacketSize (  
    spinCamera hCamera,  
    unsigned int * pMaxPacketSize )
```

Returns the largest packet size that can be safely used on the interface that device is connected to.

## See also

spinError

## Parameters

|                       |                                  |
|-----------------------|----------------------------------|
| <i>hCamera</i>        | The camera to check              |
| <i>pMaxPacketSize</i> | The maximum packet size returned |

**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.4 spinCameraEndAcquisition()**

```
SPINNAKERC_API spinCameraEndAcquisition (
    spinCamera hCamera )
```

Has a camera stop acquiring images.

**See also**

spinError

**Parameters**

|                |                                     |
|----------------|-------------------------------------|
| <i>hCamera</i> | The camera to stop acquiring images |
|----------------|-------------------------------------|

**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.5 spinCameraForceIP()**

```
SPINNAKERC_API spinCameraForceIP ( )
```

Forces the camera to be on the same subnet as its corresponding interface.

**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.6 spinCameraGetAccessMode()**

```
SPINNAKERC_API spinCameraGetAccessMode (
    spinCamera hCamera,
    spinAccessMode * pAccessMode )
```

Retrieves the access mode of a camera (as an enum, spinAccessMode)

**See also**

spinError

spinAccessMode

## Parameters

|                    |                                                                   |
|--------------------|-------------------------------------------------------------------|
| <i>hCamera</i>     | The camera of the access mode to retrieve                         |
| <i>pAccessMode</i> | The access mode enum pointer in which the access mode is returned |

## Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.7 spinCameraGetGuiXml()**

```
SPINNAKERC_API spinCameraGetGuiXml (
    spinCamera hCamera,
    char * pBuf,
    size_t * pBufLen )
```

Retrieves the GUI XML from a camera.

## See also

spinError

## Parameters

|                |                                                                                                                     |
|----------------|---------------------------------------------------------------------------------------------------------------------|
| <i>hCamera</i> | The camera of the GUI XML to retrieve                                                                               |
| <i>pBuf</i>    | The c-string character buffer in which the GUI XML is returned                                                      |
| <i>pBufLen</i> | The unsigned integer pointer in which the length of the c-string is returned; the input value is the maximum length |

## Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.8 spinCameraGetNextImage()**

```
SPINNAKERC_API spinCameraGetNextImage (
    spinCamera hCamera,
    spinImage * phImage )
```

Retrieves an image from a camera.

## See also

spinError

**Parameters**

|                |                                                         |
|----------------|---------------------------------------------------------|
| <i>hCamera</i> | The camera of the image to retrieve                     |
| <i>phImage</i> | The image handle pointer in which the image is returned |

**Returns**

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

**6.5.1.9 spinCameraGetNextImageEx()**

```
SPINNAKERC_API spinCameraGetNextImageEx (
    spinCamera hCamera,
    uint64_t grabTimeout,
    spinImage * phImage )
```

Retrieves an image from a camera; manually set the timeout in milliseconds.

**See also**

`spinError`

**Parameters**

|                    |                                                         |
|--------------------|---------------------------------------------------------|
| <i>hCamera</i>     | The camera of the image to retrieve                     |
| <i>grabTimeout</i> | The timeout value for returned an image                 |
| <i>phImage</i>     | The image handle pointer in which the image is returned |

**Returns**

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

**6.5.1.10 spinCameraGetNodeMap()**

```
SPINNAKERC_API spinCameraGetNodeMap (
    spinCamera hCamera,
    spinNodeMapHandle * phNodeMap )
```

Retrieves the GenICam nodemap from a camera.

**See also**

`spinError`



## Parameters

|                  |                                                             |
|------------------|-------------------------------------------------------------|
| <i>hCamera</i>   | The camera from which the nodemap is retrieved              |
| <i>phNodeMap</i> | The nodemap handle pointer in which the nodemap is returned |

## Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.11 spinCameraGetTLDeviceNodeMap()**

```
SPINNAKERC_API spinCameraGetTLDeviceNodeMap (  
    spinCamera hCamera,  
    spinNodeMapHandle * phNodeMap )
```

Retrieves the transport layer device nodemap from a camera.

## See also

spinError

## Parameters

|                  |                                                                       |
|------------------|-----------------------------------------------------------------------|
| <i>hCamera</i>   | The camera from which the transport layer device nodemap is retrieved |
| <i>phNodeMap</i> | The nodemap handle pointer in which the nodemap is returned           |

## Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.12 spinCameraGetTLStreamNodeMap()**

```
SPINNAKERC_API spinCameraGetTLStreamNodeMap (  
    spinCamera hCamera,  
    spinNodeMapHandle * phNodeMap )
```

Retrieves the transport layer stream nodemap from a camera.

## See also

spinError

**Parameters**

|                  |                                                                          |
|------------------|--------------------------------------------------------------------------|
| <i>hCamera</i>   | The camera from which the transport layer streaming nodemap is retrieved |
| <i>phNodeMap</i> | The nodemap handle pointer in which the nodemap is returned              |

**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.13 spinCameraGetUniqueID()**

```
SPINNAKERC_API spinCameraGetUniqueID (
    spinCamera hCamera,
    char * pBuf,
    size_t * pBufLen )
```

Retrieves a unique identifier for a camera.

**See also**

spinError

**Parameters**

|                |                                                                                                                     |
|----------------|---------------------------------------------------------------------------------------------------------------------|
| <i>hCamera</i> | The camera of the unique identifier                                                                                 |
| <i>pBuf</i>    | The c-string character buffer in which the unique identifier is returned                                            |
| <i>pBufLen</i> | The unsigned integer pointer in which the length of the c-string is returned; the input value is the maximum length |

**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.14 spinCameraInit()**

```
SPINNAKERC_API spinCameraInit (
    spinCamera hCamera )
```

Initializes a camera, allowing for much more interaction.

**See also**

spinError

## Parameters

|                |                          |
|----------------|--------------------------|
| <i>hCamera</i> | The camera to initialize |
|----------------|--------------------------|

## Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.15 spinCameraIsInitialized()**

```
SPINNAKERC_API spinCameraIsInitialized (  
    spinCamera hCamera,  
    bool8_t * pbInit )
```

Checks whether a camera is currently initialized.

## See also

spinError

## Parameters

|                |                                                                        |
|----------------|------------------------------------------------------------------------|
| <i>hCamera</i> | The camera to check                                                    |
| <i>pbInit</i>  | The boolean pointer to return whether or not the camera is initialized |

## Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.16 spinCameraIsStreaming()**

```
SPINNAKERC_API spinCameraIsStreaming (  
    spinCamera hCamera,  
    bool8_t * pbIsStreaming )
```

Checks whether a camera is currently acquiring images.

## See also

spinError

## Parameters

|                      |                                                                                |
|----------------------|--------------------------------------------------------------------------------|
| <i>hCamera</i>       | The camera to check                                                            |
| <i>pbIsStreaming</i> | The boolean pointer to return whether or not the camera is currently streaming |

**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.17 spinCamerasValid()**

```
SPINNAKERC_API spinCameraIsValid (
    spinCamera hCamera,
    bool8_t * pbValid )
```

Checks whether a camera is still valid for use.

**See also**

spinError

**Parameters**

|                |                                                                  |
|----------------|------------------------------------------------------------------|
| <i>hCamera</i> | The camera to check                                              |
| <i>pbValid</i> | The boolean pointer to return whether or not the camera is valid |

**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.18 spinCameraListAppend()**

```
SPINNAKERC_API spinCameraListAppend (
    spinCameraList hCameraListBase,
    spinCameraList hCameraListToAppend )
```

Appends all the cameras from one camera list to another.

**See also**

spinError

**Parameters**

|                            |                                      |
|----------------------------|--------------------------------------|
| <i>hCameraListBase</i>     | The camera list to receive the other |
| <i>hCameraListToAppend</i> | The camera list to add to the other  |

**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.19 spinCameraListClear()**

```
SPINNAKERC_API spinCameraListClear (  
    spinCameraList hCameraList )
```

Clears a camera list.

**See also**

spinError

**Parameters**

|                    |                          |
|--------------------|--------------------------|
| <i>hCameraList</i> | The camera list to clear |
|--------------------|--------------------------|

**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.20 spinCameraListCreateEmpty()**

```
SPINNAKERC_API spinCameraListCreateEmpty (  
    spinCameraList * phCameraList )
```

Creates an empty camera list (camera lists created this way must be destroyed)

**See also**

spinError

**Parameters**

|                     |                                                                           |
|---------------------|---------------------------------------------------------------------------|
| <i>phCameraList</i> | The camera list handle pointer in which the empty camera list is returned |
|---------------------|---------------------------------------------------------------------------|

**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

#### 6.5.1.21 spinCameraListDestroy()

```
SPINNAKERC_API spinCameraListDestroy (
    spinCameraList hCameraList )
```

Destroys a camera list.

##### See also

spinError

##### Parameters

|                    |                            |
|--------------------|----------------------------|
| <i>hCameraList</i> | The camera list to destroy |
|--------------------|----------------------------|

##### Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

#### 6.5.1.22 spinCameraListGet()

```
SPINNAKERC_API spinCameraListGet (
    spinCameraList hCameraList,
    size_t index,
    spinCamera * phCamera )
```

Retrieves a camera from a camera list using an index.

This function will return a SPINNAKER\_ERR\_INVALID\_PARAMETER error if the input index is out of range.

##### See also

spinError

##### Parameters

|                    |                                                           |
|--------------------|-----------------------------------------------------------|
| <i>hCameraList</i> | The camera list of the camera to retrieve                 |
| <i>index</i>       | The index of the camera                                   |
| <i>phCamera</i>    | The camera handle pointer in which the camera is returned |

##### Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

#### 6.5.1.23 spinCameraListGetBySerial()

```
SPINNAKERC_API spinCameraListGetBySerial (
    spinCameraList hCameraList,
    const char * pSerial,
    spinCamera * phCamera )
```

Retrieves a camera from a camera list using its serial number.

This function will return a NULL spinCamera pointer if no matching camera serial is found.

See also

spinError

Parameters

|                    |                                                           |
|--------------------|-----------------------------------------------------------|
| <i>hCameraList</i> | The camera list of the camera to retrieve                 |
| <i>serial</i>      | The serial number of the camera to retrieve               |
| <i>phCamera</i>    | The camera handle pointer in which the camera is returned |

Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

#### 6.5.1.24 spinCameraListGetSize()

```
SPINNAKERC_API spinCameraListGetSize (
    spinCameraList hCameraList,
    size_t * pSize )
```

Retrieves the number of cameras on a camera list.

See also

spinError

Parameters

|                    |                                                                         |
|--------------------|-------------------------------------------------------------------------|
| <i>hCameraList</i> | The camera list where the cameras to be counted are                     |
| <i>pSize</i>       | The unsigned integer pointer in which the number of cameras is returned |

Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

#### 6.5.1.25 spinCameraListRemove()

```
SPINNAKERC_API spinCameraListRemove (
    spinCameraList hCameraList,
    size_t index )
```

Removes a camera from a camera list using its index.

See also

spinError

Parameters

|                    |                                         |
|--------------------|-----------------------------------------|
| <i>hCameraList</i> | The camera list of the camera to remove |
| <i>index</i>       | The index of the camera to remove       |

Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

#### 6.5.1.26 spinCameraListRemoveBySerial()

```
SPINNAKERC_API spinCameraListRemoveBySerial (
    spinCameraList hCameraList,
    const char * pSerial )
```

Removes a camera from a camera list using its serial number.

See also

spinError

Parameters

|                    |                                           |
|--------------------|-------------------------------------------|
| <i>hCameraList</i> | The camera of the camera to remove        |
| <i>pSerial</i>     | The serial number of the camera to remove |

Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

#### 6.5.1.27 spinCameraReadPort()

```
SPINNAKERC_API spinCameraReadPort (
    spinCamera hCamera,
```



```
uint64_t iAddress,  
void * pBuffer,  
size_t iSize )
```

#### 6.5.1.28 spinCameraRegisterDeviceEventHandler()

```
SPINNAKERC_API spinCameraRegisterDeviceEventHandler (   
    spinCamera hCamera,  
    spinDeviceEventHandler hDeviceEventHandler )
```

Registers a universal device event handler (every device event type) to a camera.

See also

spinError

##### Parameters

|                            |                                                                    |
|----------------------------|--------------------------------------------------------------------|
| <i>hCamera</i>             | The camera on which to register the universal device event handler |
| <i>hDeviceEventHandler</i> | The device event handler to register                               |

##### Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

#### 6.5.1.29 spinCameraRegisterDeviceEventHandlerEx()

```
SPINNAKERC_API spinCameraRegisterDeviceEventHandlerEx (   
    spinCamera hCamera,  
    spinDeviceEventHandler hDeviceEventHandler,  
    const char * pName )
```

Registers a specific device event handler (only one device event type) to a camera.

See also

spinError

##### Parameters

|                            |                                                                   |
|----------------------------|-------------------------------------------------------------------|
| <i>hCamera</i>             | The camera on which to register the specific device event handler |
| <i>hDeviceEventHandler</i> | The device event handler to register                              |
| <i>pName</i>               | The name of the device event handler to register                  |

**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.30 spinCameraRegisterImageEventHandler()**

```
SPINNAKERC_API spinCameraRegisterImageEventHandler (
    spinCamera hCamera,
    spinImageEventHandler hImageEventHandler )
```

Registers an image event handler to a camera.

**See also**

spinError

**Parameters**

|                           |                                                         |
|---------------------------|---------------------------------------------------------|
| <i>hCamera</i>            | The camera on which to register the image event handler |
| <i>hImageEventHandler</i> | The image event handler to register                     |

**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.31 spinCameraRelease()**

```
SPINNAKERC_API spinCameraRelease (
    spinCamera hCamera )
```

Releases a camera.

**See also**

spinError

**Parameters**

|                |                       |
|----------------|-----------------------|
| <i>hCamera</i> | The camera to release |
|----------------|-----------------------|

**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

#### 6.5.1.32 spinCameraUnregisterDeviceEventHandler()

```
SPINNAKERC_API spinCameraUnregisterDeviceEventHandler (
    spinCamera hCamera,
    spinDeviceEventHandler hDeviceEventHandler )
```

Unregisters a device event handler from a camera.

##### See also

spinError

##### Parameters

|                            |                                                              |
|----------------------------|--------------------------------------------------------------|
| <i>hCamera</i>             | The camera from which to unregister the device event handler |
| <i>hDeviceEventHandler</i> | The device event handler to unregister                       |

##### Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

#### 6.5.1.33 spinCameraUnregisterImageEventHandler()

```
SPINNAKERC_API spinCameraUnregisterImageEventHandler (
    spinCamera hCamera,
    spinImageEventHandler hImageEventHandler )
```

Unregisters an image event handler from a camera.

##### See also

spinError

##### Parameters

|                           |                                                             |
|---------------------------|-------------------------------------------------------------|
| <i>hCamera</i>            | The camera from which to unregister the image event handler |
| <i>hImageEventHandler</i> | The image event handler to unregister                       |

##### Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

#### 6.5.1.34 spinCameraWritePort()

```
SPINNAKERC_API spinCameraWritePort (
    spinCamera hCamera,
```

```
uint64_t iAddress,
void * pBuffer,
size_t iSize )
```

#### 6.5.1.35 spinDeviceArrivalEventHandlerCreate()

```
SPINNAKERC_API spinDeviceArrivalEventHandlerCreate (
    spinDeviceArrivalEventHandler * phDeviceArrivalEventHandler,
    spinArrivalEventFunction pFunction,
    void * pUserData )
```

Creates a device arrival event handler.

See also

spinError

##### Parameters

|                                    |                                                                                                                               |
|------------------------------------|-------------------------------------------------------------------------------------------------------------------------------|
| <i>phDeviceArrivalEventHandler</i> | The device arrival event handler pointer in which the device arrival event context is created                                 |
| <i>pFunction</i>                   | The function to be called at device event occurrences; signature to match: void(<em>spinArrivalEventFunction)(void pUserData) |
| <i>pUserData</i>                   | Properties that can be passed into the event function                                                                         |

##### Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

#### 6.5.1.36 spinDeviceArrivalEventHandlerDestroy()

```
SPINNAKERC_API spinDeviceArrivalEventHandlerDestroy (
    spinDeviceArrivalEventHandler hDeviceArrivalEventHandler )
```

Destroys a device arrival event handler.

See also

spinError

##### Parameters

|                                   |                                             |
|-----------------------------------|---------------------------------------------|
| <i>hDeviceArrivalEventHandler</i> | The device arrival event handler to destroy |
|-----------------------------------|---------------------------------------------|

**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.37 spinDeviceEventGetId()**

```
SPINNAKERC_API spinDeviceEventGetId (
    spinDeviceEventData hDeviceEventData,
    uint64_t * pEventId )
```

Retrieves the event ID of a device event.

**See also**

spinError

**Parameters**

|                         |                                                                |
|-------------------------|----------------------------------------------------------------|
| <i>hDeviceEventData</i> | The log event data received from the log event                 |
| <i>pEventId</i>         | The unsigned integer pointer in which the event ID is returned |

**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.38 spinDeviceEventGetName()**

```
SPINNAKERC_API spinDeviceEventGetName (
    spinDeviceEventData hDeviceEventData,
    char * pBuf,
    size_t * pBufLen )
```

Retrieves the event name of a device event.

**See also**

spinError

**Parameters**

|                         |                                                                                                                     |
|-------------------------|---------------------------------------------------------------------------------------------------------------------|
| <i>hDeviceEventData</i> | The log event data received from the log event                                                                      |
| <i>pBuf</i>             | The c-string character buffer in which the name of the device event is returned                                     |
| <i>pBufLen</i>          | The unsigned integer pointer in which the length of the c-string is returned; the input value is the maximum length |

**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.39 spinDeviceEventGetPayloadData()**

```
SPINNAKERC_API spinDeviceEventGetPayloadData (
    spinDeviceEventData hDeviceEventData,
    const uint8_t * pBuf,
    size_t * pBufSize )
```

Retrieves the payload data of a device event.

**See also**

spinError

**Parameters**

|                         |                                                                           |
|-------------------------|---------------------------------------------------------------------------|
| <i>hDeviceEventData</i> | The log event data received from the log event                            |
| <i>pBuf</i>             | The unsigned integer pointer in which the event payload is returned       |
| <i>pBufSize</i>         | The unsigned integer pointer in which the size of the payload is returned |

**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.40 spinDeviceEventGetPayloadDataSize()**

```
SPINNAKERC_API spinDeviceEventGetPayloadDataSize (
    spinDeviceEventData hDeviceEventData,
    size_t * pBufSize )
```

Retrieves the payload data size of a device event.

**See also**

spinError

**Parameters**

|                         |                                                                           |
|-------------------------|---------------------------------------------------------------------------|
| <i>hDeviceEventData</i> | The log event data received from the log event                            |
| <i>pBufSize</i>         | The unsigned integer pointer in which the size of the payload is returned |

**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.41 spinDeviceEventHandlerCreate()**

```
SPINNAKERC_API spinDeviceEventHandlerCreate (
    spinDeviceEventHandler * phDeviceEventHandler,
    spinDeviceEventFunction pFunction,
    void * pUserData )
```

Creates a device event handler.

**See also**

spinError

**Parameters**

|                             |                                                                                                                                                                                                  |
|-----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>phDeviceEventHandler</i> | The device event handler pointer in which the device event context is created                                                                                                                    |
| <i>pFunction</i>            | The function to be called at device event occurrences; signature to match:<br>void(<em>spinDeviceEventFunction)(const spinDeviceEventData hEventData,<br>const char pEventName, void* pUserData) |
| <i>pUserData</i>            | Properties that can be passed into the event function                                                                                                                                            |

**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.42 spinDeviceEventHandlerDestroy()**

```
SPINNAKERC_API spinDeviceEventHandlerDestroy (
    spinDeviceEventHandler hDeviceEventHandler )
```

Destroys a device event handler.

**See also**

spinError

**Parameters**

|                            |                                     |
|----------------------------|-------------------------------------|
| <i>hDeviceEventHandler</i> | The device event handler to destroy |
|----------------------------|-------------------------------------|

**Returns**

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

**6.5.1.43 spinDeviceRemovalEventHandlerCreate()**

```
SPINNAKERC_API spinDeviceRemovalEventHandlerCreate (
    spinDeviceRemovalEventHandler * phDeviceRemovalEventHandler,
    spinRemovalEventFunction pFunction,
    void * pUserData )
```

Creates a device removal event handler.

**See also**

`spinError`

**Parameters**

|                                    |                                                                                                                                                                               |
|------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>phDeviceRemovalEventHandler</i> | The device removal event handler pointer in which the device removal event context is created                                                                                 |
| <i>pFunction</i>                   | The function to be called at device event occurrences; signature to match: <code>void(&lt;em&gt;spinRemovalEventFunction)(uint64_t deviceSerialNumber, void pUserData)</code> |
| <i>pUserData</i>                   | Properties that can be passed into the event function                                                                                                                         |

**Returns**

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

**6.5.1.44 spinDeviceRemovalEventHandlerDestroy()**

```
SPINNAKERC_API spinDeviceRemovalEventHandlerDestroy (
    spinDeviceRemovalEventHandler hDeviceRemovalEventHandler )
```

Destroys a device removal event handler.

**See also**

`spinError`

**Parameters**

|                                   |                                             |
|-----------------------------------|---------------------------------------------|
| <i>hDeviceRemovalEventHandler</i> | The device removal event handler to destroy |
|-----------------------------------|---------------------------------------------|



**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.45 spinErrorGetLast()**

```
SPINNAKERC_API spinErrorGetLast (
    spinError * pError )
```

Retrieves the error code of the last error.

**See also**

spinError

**Parameters**

|               |                                                               |
|---------------|---------------------------------------------------------------|
| <i>pError</i> | The error enum pointer in which the error message is returned |
|---------------|---------------------------------------------------------------|

**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.46 spinErrorGetLastBuildDate()**

```
SPINNAKERC_API spinErrorGetLastBuildDate (
    char * pBuf,
    size_t * pBufLen )
```

Retrieves the build date of the last error.

**See also**

spinError

**Parameters**

|                |                                                                                                                     |
|----------------|---------------------------------------------------------------------------------------------------------------------|
| <i>pBuf</i>    | The c-string character buffer in which the build date is returned                                                   |
| <i>pBufLen</i> | The unsigned integer pointer in which the length of the c-string is returned; the input value is the maximum length |

**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

#### 6.5.1.47 spinErrorGetLastBuildTime()

```
SPINNAKERC_API spinErrorGetLastBuildTime (
    char * pBuf,
    size_t * pBufLen )
```

Retrieves the build time of the last error.

See also

spinError

##### Parameters

|                |                                                                                                                     |
|----------------|---------------------------------------------------------------------------------------------------------------------|
| <i>pBuf</i>    | The c-string character buffer in which the build time is returned                                                   |
| <i>pBufLen</i> | The unsigned integer pointer in which the length of the c-string is returned; the input value is the maximum length |

##### Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

#### 6.5.1.48 spinErrorGetLastFileName()

```
SPINNAKERC_API spinErrorGetLastFileName (
    char * pBuf,
    size_t * pBufLen )
```

Retrieves the filename of the last error.

See also

spinError

##### Parameters

|                |                                                                                                                     |
|----------------|---------------------------------------------------------------------------------------------------------------------|
| <i>pBuf</i>    | The c-string character buffer in which the file name is returned                                                    |
| <i>pBufLen</i> | The unsigned integer pointer in which the length of the c-string is returned; the input value is the maximum length |

##### Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

#### 6.5.1.49 spinErrorGetLastFullMessage()

```
SPINNAKERC_API spinErrorGetLastFullMessage (
    char * pBuf,
    size_t * pBufLen )
```

Retrieves the full error message of the last error.

See also

spinError

##### Parameters

|                |                                                                                                                     |
|----------------|---------------------------------------------------------------------------------------------------------------------|
| <i>pBuf</i>    | The c-string character buffer in which the full error message is returned                                           |
| <i>pBufLen</i> | The unsigned integer pointer in which the length of the c-string is returned; the input value is the maximum length |

##### Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

#### 6.5.1.50 spinErrorGetLastFunctionName()

```
SPINNAKERC_API spinErrorGetLastFunctionName (
    char * pBuf,
    size_t * pBufLen )
```

Retrieves the function name of the last error.

See also

spinError

##### Parameters

|                |                                                                                                                     |
|----------------|---------------------------------------------------------------------------------------------------------------------|
| <i>pBuf</i>    | The c-string character buffer in which the function name is returned                                                |
| <i>pBufLen</i> | The unsigned integer pointer in which the length of the c-string is returned; the input value is the maximum length |

##### Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

#### 6.5.1.51 spinErrorGetLastLineNumber()

```
SPINNAKERC_API spinErrorGetLastLineNumber (
    int64_t * pLineNumber )
```

Retrieves the line number of the last error.

See also

spinError

##### Parameters

|                |                                                                                                                     |
|----------------|---------------------------------------------------------------------------------------------------------------------|
| <i>pBuf</i>    | The c-string character buffer in which the line number is returned                                                  |
| <i>pBufLen</i> | The unsigned integer pointer in which the length of the c-string is returned; the input value is the maximum length |

##### Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

#### 6.5.1.52 spinErrorGetLastMessage()

```
SPINNAKERC_API spinErrorGetLastMessage (
    char * pBuf,
    size_t * pBufLen )
```

Retrieves the error message of the last error.

See also

spinError

##### Parameters

|                |                                                                                                                     |
|----------------|---------------------------------------------------------------------------------------------------------------------|
| <i>pBuf</i>    | The c-string character buffer in which the error message is returned                                                |
| <i>pBufLen</i> | The unsigned integer pointer in which the length of the c-string is returned; the input value is the maximum length |

##### Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

#### 6.5.1.53 spinImageCalculateStatistics()

```
SPINNAKERC_API spinImageCalculateStatistics (
    spinImage hImage,
    const spinImageStatistics hStatistics )
```

Calculates the image statistics of an image.

See also

spinError

Parameters

|                    |                                                                              |
|--------------------|------------------------------------------------------------------------------|
| <i>hImage</i>      | The image to be saved                                                        |
| <i>hStatistics</i> | The image statistics context in which the calculated statistics are returned |

Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

#### 6.5.1.54 spinImageCheckCRC()

```
SPINNAKERC_API spinImageCheckCRC (
    spinImage hImage,
    bool8_t * pbCheckCRC )
```

Checks whether the CRC of an image is correct.

See also

spinError

Parameters

|                   |                                                            |
|-------------------|------------------------------------------------------------|
| <i>hImage</i>     | The image to be saved                                      |
| <i>pbCheckCRC</i> | The boolean pointer to return whether the image CRC passes |

Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

#### 6.5.1.55 spinImageChunkDataGetFloatValue()

```
SPINNAKERC_API spinImageChunkDataGetFloatValue (
    spinImage hImage,
```

```
const char * pName,  
double * pValue )
```

#### 6.5.1.56 spinImageChunkDataGetIntValue()

```
SPINNAKERC_API spinImageChunkDataGetIntValue (   
    spinImage hImage,  
    const char * pName,  
    int64_t * pValue )
```

#### 6.5.1.57 spinImageConvert()

```
SPINNAKERC_API spinImageConvert (   
    spinImage hSrcImage,  
    spinPixelFormatEnums pixelFormat,  
    spinImage hDestImage )
```

Converts the pixel format of one image into a new image.

See also

[spinError](#)

##### Parameters

|                    |                                                                   |
|--------------------|-------------------------------------------------------------------|
| <i>hSrcImage</i>   | The image to be converted                                         |
| <i>pixelFormat</i> | The pixel format to be converted to                               |
| <i>hDestImage</i>  | The image handle pointer in which the converted image is returned |

##### Returns

[spinError](#) The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

#### 6.5.1.58 spinImageConvertEx()

```
SPINNAKERC_API spinImageConvertEx (   
    spinImage hSrcImage,  
    spinPixelFormatEnums pixelFormat,  
    spinColorProcessingAlgorithm algorithm,  
    spinImage hDestImage )
```

Converts the pixel format and color processing algorithm of one image into a new image.

See also

[spinError](#)

## Parameters

|                    |                                                                   |
|--------------------|-------------------------------------------------------------------|
| <i>hSrcImage</i>   | The image to be converted                                         |
| <i>pixelFormat</i> | The pixel format to be converted to                               |
| <i>algorithm</i>   | The color processing algorithm to use for conversion              |
| <i>hDestImage</i>  | The image handle pointer in which the converted image is returned |

## Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.59 spinImageCreate()**

```
SPINNAKERC_API spinImageCreate (
    spinImage hSrcImage,
    spinImage * phDestImage )
```

Creates an image from another; images created this way must be destroyed.

## See also

spinError

## Parameters

|                    |                                                     |
|--------------------|-----------------------------------------------------|
| <i>hSrcImage</i>   | The image to be copied                              |
| <i>phDestImage</i> | The image handle pointer of the image to be created |

## Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.60 spinImageCreateEmpty()**

```
SPINNAKERC_API spinImageCreateEmpty (
    spinImage * phImage )
```

Creates an empty image; images created this way must be destroyed.

## See also

spinError

## Parameters

|                |                                                               |
|----------------|---------------------------------------------------------------|
| <i>phImage</i> | The image handle pointer in which the empty image is returned |
|----------------|---------------------------------------------------------------|

## Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

**6.5.1.61 spinImageCreateEx()**

```
SPINNAKERC_API spinImageCreateEx (
    spinImage * phImage,
    size_t width,
    size_t height,
    size_t offsetX,
    size_t offsetY,
    spinPixelFormatEnums pixelFormat,
    void * pData )
```

Creates an image with some set properties; images created this way must be destroyed.

## See also

`spinError`

## Parameters

|                    |                                                         |
|--------------------|---------------------------------------------------------|
| <i>phImage</i>     | The image handle pointer in which the image is returned |
| <i>width</i>       | The width to set                                        |
| <i>height</i>      | The height to set                                       |
| <i>offsetX</i>     | The offset along the X axis to set                      |
| <i>offsetY</i>     | The offset along the Y axis to set                      |
| <i>pixelFormat</i> | The pixel format to set                                 |
| <i>pData</i>       | The image data to set; can be set to null               |

## Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

**6.5.1.62 spinImageDeepCopy()**

```
SPINNAKERC_API spinImageDeepCopy (
    spinImage hSrcImage,
    spinImage hDestImage )
```

Creates a deep copy of an image (the destination image must be created as an empty image prior to the deep copy)



## See also

spinError

## Parameters

|                   |                                               |
|-------------------|-----------------------------------------------|
| <i>hSrcImage</i>  | The image to be copied                        |
| <i>hDestImage</i> | The image handle in which the image is copied |

## Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.63 spinImageDestroy()**

```
SPINNAKERC_API spinImageDestroy (  
    spinImage hImage )
```

Destroys an image.

## See also

spinError

## Parameters

|               |                      |
|---------------|----------------------|
| <i>hImage</i> | The image to destroy |
|---------------|----------------------|

## Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.64 spinImageEventHandlerCreate()**

```
SPINNAKERC_API spinImageEventHandlerCreate (  
    spinImageEventHandler * phImageEventHandler,  
    spinImageEventFunction pFunction,  
    void * pUserData )
```

Creates an image event handler.

## See also

spinError

## Parameters

|                            |                                                                                                                                                       |
|----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>phImageEventHandler</i> | The image event handler pointer in which the image event context is created                                                                           |
| <i>pFunction</i>           | The function to be called at image event occurrences; signature to match:<br>void(<em>spinImageEventFunction)(const spinImage hImage, void pUserData) |
| <i>pUserData</i>           | Properties that can be passed into the event function                                                                                                 |

## Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.65 spinImageEventHandlerDestroy()**

```
SPINNAKERC_API spinImageEventHandlerDestroy (
    spinImageEventHandler hImageEventHandler )
```

Destroys an image event handler.

## See also

spinError

## Parameters

|                           |                                    |
|---------------------------|------------------------------------|
| <i>hImageEventHandler</i> | The image event handler to destroy |
|---------------------------|------------------------------------|

## Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.66 spinImageGetBitsPerPixel()**

```
SPINNAKERC_API spinImageGetBitsPerPixel (
    spinImage hImage,
    size_t * pBitsPerPixel )
```

Retrieves the number of bits per pixel of an image.

## See also

spinError

## Parameters

|                      |                                                                                |
|----------------------|--------------------------------------------------------------------------------|
| <i>hImage</i>        | The image to be saved                                                          |
| <i>pBitsPerPixel</i> | The unsigned integer pointer in which the number of bits per pixel is returned |

## Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.67 spinImageGetBufferSize()**

```
SPINNAKERC_API spinImageGetBufferSize (
    spinImage hImage,
    size_t * pSize )
```

Retrieves the buffer size of an image.

## See also

spinError

## Parameters

|               |                                                                              |
|---------------|------------------------------------------------------------------------------|
| <i>hImage</i> | The image of image data buffer to retrieve                                   |
| <i>pSize</i>  | The unsigned integer pointer in which the size of the image data if returned |

## Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.68 spinImageGetChunkLayoutID()**

```
SPINNAKERC_API spinImageGetChunkLayoutID (
    spinImage hImage,
    uint64_t * pId )
```

Retrieves the chunk layout ID of an image.

## See also

spinError

**Parameters**

|               |                                                                       |
|---------------|-----------------------------------------------------------------------|
| <i>hImage</i> | The image to be saved                                                 |
| <i>pId</i>    | The unsigned integer pointer in which the chunk layout ID is returned |

**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.69 spinImageGetColorProcessing()**

```
SPINNAKERC_API spinImageGetColorProcessing (
    spinImage hImage,
    spinColorProcessingAlgorithm * pAlgorithm )
```

Retrieves the color processing algorithm of a specific image.

**See also**

spinError

**Parameters**

|                   |                                                                                            |
|-------------------|--------------------------------------------------------------------------------------------|
| <i>hImage</i>     | The image of the color processing algorithm to retrieve                                    |
| <i>pAlgorithm</i> | The color processing algorithm pointer in which the color processing algorithm is returned |

**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.70 spinImageGetData()**

```
SPINNAKERC_API spinImageGetData (
    spinImage hImage,
    void ** ppData )
```

Retrieves the image data of an image.

**See also**

spinError

## Parameters

|               |                                                                      |
|---------------|----------------------------------------------------------------------|
| <i>hImage</i> | The image of the image data to retrieve                              |
| <i>ppData</i> | The pointer to the void pointer in which the image data is retrieved |

## Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.71 spinImageGetDefaultColorProcessing()**

```
SPINNAKERC_API spinImageGetDefaultColorProcessing (
    spinColorProcessingAlgorithm * pAlgorithm )
```

Retrieves the default color processing algorithm.

## See also

spinError

## Parameters

|                   |                                                                                                 |
|-------------------|-------------------------------------------------------------------------------------------------|
| <i>pAlgorithm</i> | The color processing algorithm enum pointer in which the color processing algorithm is returned |
|-------------------|-------------------------------------------------------------------------------------------------|

## Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.72 spinImageGetFrameID()**

```
SPINNAKERC_API spinImageGetFrameID (
    spinImage hImage,
    uint64_t * pFrameID )
```

Retrieves the frame ID of an image.

## See also

spinError

## Parameters

|                 |                                                                |
|-----------------|----------------------------------------------------------------|
| <i>hImage</i>   | The image of the frame ID to retrieve                          |
| <i>pFrameID</i> | The unsigned integer pointer in which the frame ID is returned |

**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.73 spinImageGetHeight()**

```
SPINNAKERC_API spinImageGetHeight (
    spinImage hImage,
    size_t * pHeight )
```

Retrieves the height of an image.

**See also**

spinError

**Parameters**

|                |                                                              |
|----------------|--------------------------------------------------------------|
| <i>hImage</i>  | The image of the height to retrieve                          |
| <i>pHeight</i> | The unsigned integer pointer in which the height is returned |

**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.74 spinImageGetID()**

```
SPINNAKERC_API spinImageGetID (
    spinImage hImage,
    uint64_t * pId )
```

Retrieves the ID of an image.

**See also**

spinError

**Parameters**

|               |                                                          |
|---------------|----------------------------------------------------------|
| <i>hImage</i> | The image of the ID to retrieve                          |
| <i>pId</i>    | The unsigned integer pointer in which the ID is returned |

**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.75 spinImageGetOffsetX()**

```
SPINNAKERC_API spinImageGetOffsetX (  
    spinImage hImage,  
    size_t * pOffsetX )
```

Retrieves the offset of an image along its X axis.

**See also**

spinError

**Parameters**

|                 |                                                                               |
|-----------------|-------------------------------------------------------------------------------|
| <i>hImage</i>   | The image of the offset along the X axis to retrieve                          |
| <i>pOffsetX</i> | The unsigned integer pointer in which the offset along the X axis is returned |

**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.76 spinImageGetOffsetY()**

```
SPINNAKERC_API spinImageGetOffsetY (  
    spinImage hImage,  
    size_t * pOffsetY )
```

Retrieves the offset of an image along its Y axis.

**See also**

spinError

**Parameters**

|                 |                                                                               |
|-----------------|-------------------------------------------------------------------------------|
| <i>hImage</i>   | The image of the offset along the Y axis to retrieve                          |
| <i>pOffsetY</i> | The unsigned integer pointer in which the offset along the Y axis is returned |

**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.77 spinImageGetPaddingX()**

```
SPINNAKERC_API spinImageGetPaddingX (
    spinImage hImage,
    size_t * pPaddingX )
```

Retrieves the padding of an image along its X axis.

**See also**

spinError

**Parameters**

|                  |                                                                                |
|------------------|--------------------------------------------------------------------------------|
| <i>hImage</i>    | The image of the padding along the X axis to retrieve                          |
| <i>pPaddingX</i> | The unsigned integer pointer in which the padding along the X axis is returned |

**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.78 spinImageGetPaddingY()**

```
SPINNAKERC_API spinImageGetPaddingY (
    spinImage hImage,
    size_t * pPaddingY )
```

Retrieves the padding of an image along its Y axis.

**See also**

spinError

**Parameters**

|                  |                                                                                |
|------------------|--------------------------------------------------------------------------------|
| <i>hImage</i>    | The image of the padding along the Y axis to retrieve                          |
| <i>pPaddingY</i> | The unsigned integer pointer in which the padding along the Y axis is returned |



**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.79 spinImageGetPayloadType()**

```
SPINNAKERC_API spinImageGetPayloadType (
    spinImage hImage,
    size_t * pPayloadType )
```

Retrieves the payload type of an image (as an enum, spinPayloadTypeIn folds)

**See also**

spinError  
spinPayloadTypeIn folds

**Parameters**

|                     |                                                                     |
|---------------------|---------------------------------------------------------------------|
| <i>hImage</i>       | The image of the payload type to retrieve                           |
| <i>pPayloadType</i> | The payload type enum pointer in which the payload type is returned |

**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.80 spinImageGetPixelFormat()**

```
SPINNAKERC_API spinImageGetPixelFormat (
    spinImage hImage,
    spinPixelFormatEnums * pPixelFormat )
```

Retrieves the pixel format of an image (as an enum, spinPixelFormatEnums)

**See also**

spinError  
spinPixelFormatEnums

**Parameters**

|                     |                                                                     |
|---------------------|---------------------------------------------------------------------|
| <i>hImage</i>       | The image of the pixel format to retrieve                           |
| <i>pPixelFormat</i> | The pixel format enum pointer in which the pixel format is returned |

**Returns**

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

**6.5.1.81 spinImageGetPixelFormatName()**

```
SPINNAKERC_API spinImageGetPixelFormatName (
    spinImage hImage,
    char * pBuf,
    size_t * pBufLen )
```

Retrieves the pixel format of an image (as a symbolic)

**See also**

`spinError`

**Parameters**

|                |                                                                                                                     |
|----------------|---------------------------------------------------------------------------------------------------------------------|
| <i>hImage</i>  | The image of the pixel format to retrieve                                                                           |
| <i>pBuf</i>    | The c-string character buffer in which the pixel format symbolic is returned                                        |
| <i>pBufLen</i> | The unsigned integer pointer in which the length of the c-string is returned; the input value is the maximum length |

**Returns**

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

**6.5.1.82 spinImageGetPrivateData()**

```
SPINNAKERC_API spinImageGetPrivateData (
    spinImage hImage,
    void ** ppData )
```

Retrieves the private data of an image.

**See also**

`spinError`

**Parameters**

|               |                                                                              |
|---------------|------------------------------------------------------------------------------|
| <i>hImage</i> | The image of the private image data to retrieve                              |
| <i>ppData</i> | The pointer to the void pointer in which the private image data is retrieved |

**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.83 spinImageGetSize()**

```
SPINNAKERC_API spinImageGetSize (
    spinImage hImage,
    size_t * pImageSize )
```

Retrieves the size of an image.

**See also**

spinError

**Parameters**

|                   |                                                                         |
|-------------------|-------------------------------------------------------------------------|
| <i>hImage</i>     | The image to be saved                                                   |
| <i>pImageSize</i> | The unsigned integer pointer in which the size of the image is returned |

**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.84 spinImageGetStatus()**

```
SPINNAKERC_API spinImageGetStatus (
    spinImage hImage,
    spinImageStatus * pStatus )
```

Retrieves the image status of an image.

**See also**

spinError

**Parameters**

|                |                                                               |
|----------------|---------------------------------------------------------------|
| <i>hImage</i>  | The image to be saved                                         |
| <i>pStatus</i> | The status enum pointer in which the image status is returned |

**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.85 spinImageGetStatusDescription()**

```
SPINNAKERC_API spinImageGetStatusDescription (
    spinImageStatus status,
    char * pBuf,
    size_t * pBufLen )
```

Retrieves the description of image status.

**See also**

spinError

**Parameters**

|                |                                                                                                                                                                                   |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>status</i>  | The status enum                                                                                                                                                                   |
| <i>pBuf</i>    | The c-string character buffer in which the explanation of image status enum is returned                                                                                           |
| <i>pBufLen</i> | The unsigned integer pointer in which the length of the c-string is returned; the input value is the maximum length; if pBuf is NULL, minimum length of string buffer is returned |

**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.86 spinImageGetStride()**

```
SPINNAKERC_API spinImageGetStride (
    spinImage hImage,
    size_t * pStride )
```

Retrieves the stride of an image.

**See also**

spinError

**Parameters**

|                |                                                              |
|----------------|--------------------------------------------------------------|
| <i>hImage</i>  | The image to be saved                                        |
| <i>pStride</i> | The unsigned integer pointer in which the stride is returned |

**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.87 spinImageGetTimeStamp()**

```
SPINNAKERC_API spinImageGetTimeStamp (
    spinImage hImage,
    uint64_t * pTimeStamp )
```

Retrieves the timestamp of an image.

**See also**

spinError

**Parameters**

|                   |                                                                 |
|-------------------|-----------------------------------------------------------------|
| <i>hImage</i>     | The image of the timestamp to retrieve                          |
| <i>pTimeStamp</i> | The unsigned integer pointer om which the timestamp is returned |

**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.88 spinImageGetTLPayloadType()**

```
SPINNAKERC_API spinImageGetTLPayloadType (
    spinImage hImage,
    spinPayloadTypeInfoIDs * pPayloadType )
```

Retrieves the transport layer payload type of an image (as an enum, spinPayloadTypeInfolDs)

**See also**

spinError

spinPayloadTypeInfolDs

**Parameters**

|                     |                                                                        |
|---------------------|------------------------------------------------------------------------|
| <i>hImage</i>       | The image of the TL payload type to retrieve                           |
| <i>pPayloadType</i> | The payload type enum pointer in which the TL payload type is returned |

**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.89 spinImageGetTLPixelFormat()**

```
SPINNAKERC_API spinImageGetTLPixelFormat (
    spinImage hImage,
    uint64_t * pPixelFormat )
```

Retrieves the transport layer pixel format of an image (as an unsigned integer)

**See also**

spinError

**Parameters**

|                     |                                                                       |
|---------------------|-----------------------------------------------------------------------|
| <i>hImage</i>       | The image of the TL pixel format to retrieve                          |
| <i>pPixelFormat</i> | The unsigned integer pointer in which the TL pixel format is returned |

**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.90 spinImageGetTLPixelFormatNamespace()**

```
SPINNAKERC_API spinImageGetTLPixelFormatNamespace (
    spinImage hImage,
    spinPixelFormatNamespaceID * pPixelFormatNamespace )
```

Retrieves the transport layer pixel format namespace of an image (as an enum, spinPixelFormatNamespaceID)

**See also**

spinError

spinPixelFormatNamespaceID

**Parameters**

|                              |                                                                                    |
|------------------------------|------------------------------------------------------------------------------------|
| <i>hImage</i>                | The image of the TL pixel format namespace to retrieve                             |
| <i>pPixelFormatNamespace</i> | The pixel format namespace pointer in which the pixel format namespace is returned |

**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.91 spinImageGetValidPayloadSize()**

```
SPINNAKERC_API spinImageGetValidPayloadSize (
    spinImage hImage,
    size_t * pSize )
```

Retrieves the valid payload size of an image.

**See also**

spinError

**Parameters**

|               |                                                                                 |
|---------------|---------------------------------------------------------------------------------|
| <i>hImage</i> | The image of the payload size to retrieve                                       |
| <i>pSize</i>  | The unsigned integer pointer in which the size of the valid payload is returned |

**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.92 spinImageGetWidth()**

```
SPINNAKERC_API spinImageGetWidth (
    spinImage hImage,
    size_t * pWidth )
```

Retrieves the width of an image.

**See also**

spinError

**Parameters**

|               |                                                             |
|---------------|-------------------------------------------------------------|
| <i>hImage</i> | The image of the width to retrieve                          |
| <i>pWidth</i> | The unsigned integer pointer in which the width is returned |

**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.93 spinImageHasCRC()**

```
SPINNAKERC_API spinImageHasCRC (
    spinImage hImage,
    bool8_t * pbHasCRC )
```

Checks whether an image has CRC.

**See also**

spinError

**Parameters**

|                 |                                                                   |
|-----------------|-------------------------------------------------------------------|
| <i>hImage</i>   | The image to be saved                                             |
| <i>pbHasCRC</i> | The boolean pointer to return whether the image has CRC available |

**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.94 spinImageIsIncomplete()**

```
SPINNAKERC_API spinImageIsIncomplete (
    spinImage hImage,
    bool8_t * pbIsIncomplete )
```

Checks whether an image is incomplete.

**See also**

spinError

**Parameters**

|                       |                                                                      |
|-----------------------|----------------------------------------------------------------------|
| <i>hImage</i>         | The image to check                                                   |
| <i>pbIsIncomplete</i> | The boolean pointer to return whether or not the image is incomplete |



**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.95 spinImageRelease()**

```
SPINNAKERC_API spinImageRelease (
    spinImage hImage )
```

Releases an image.

**See also**

spinError

**Parameters**

|               |                       |
|---------------|-----------------------|
| <i>hImage</i> | The image to be saved |
|---------------|-----------------------|

**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.96 spinImageReset()**

```
SPINNAKERC_API spinImageReset (
    spinImage hImage,
    size_t width,
    size_t height,
    size_t offsetX,
    size_t offsetY,
    spinPixelFormatEnums pixelFormat )
```

Resets an image with some set properties.

**See also**

spinError

**Parameters**

|                    |                                            |
|--------------------|--------------------------------------------|
| <i>hImage</i>      | The image to be reset                      |
| <i>width</i>       | The width to be reset to                   |
| <i>height</i>      | The height to be reset to                  |
| <i>offsetX</i>     | The offset to be reset to along the X axis |
| <i>offsetY</i>     | The offset to be reset to along the Y axis |
| <i>pixelFormat</i> | The pixel format to be reset to            |

**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.97 spinImageResetEx()**

```
SPINNAKERC_API spinImageResetEx (
    spinImage hImage,
    size_t width,
    size_t height,
    size_t offsetX,
    size_t offsetY,
    spinPixelFormatEnums pixelFormat,
    void * pData )
```

Resets an image with some set properties and image data.

**See also**

spinError

**Parameters**

|                    |                                            |
|--------------------|--------------------------------------------|
| <i>hImage</i>      | The image to reset                         |
| <i>width</i>       | The width to be reset to                   |
| <i>height</i>      | The height to be reset to                  |
| <i>offsetX</i>     | The offset to be reset to along the X axis |
| <i>offsetY</i>     | The offset to be reset to along the Y axis |
| <i>pixelFormat</i> | The pixel format to be reset to            |
| <i>pData</i>       | The image data to reset to                 |

**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.98 spinImageSave()**

```
SPINNAKERC_API spinImageSave (
    spinImage hImage,
    const char * pFilename,
    spinImageFileFormat format )
```

Saves an image using a specified file format (using an enum, spinImageFileFormat)

**See also**

spinError

spinImageFileFormat

## Parameters

|                  |                                                                                                                                                  |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>hImage</i>    | The image to be saved                                                                                                                            |
| <i>pFilename</i> | The filename to use to save the image (with or without the appropriate file extension) @Param<br>format The file format to use to save the image |

## Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.99 spinImageSaveBmp()**

```
SPINNAKER_API spinImageSaveBmp (  
    spinImage hImage,  
    const char * pFilename,  
    const spinBMPOption * pOption )
```

Saves an image as a BMP image.

## See also

spinError

## Parameters

|                  |                                                                                        |
|------------------|----------------------------------------------------------------------------------------|
| <i>hImage</i>    | The image to be saved                                                                  |
| <i>pFilename</i> | The filename to use to save the image (with or without the appropriate file extension) |
| <i>pOption</i>   | The image options related to saving as BMP; includes whether to save as indexed 8-bit  |

## Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.100 spinImageSaveFromExt()**

```
SPINNAKER_API spinImageSaveFromExt (  
    spinImage hImage,  
    const char * pFilename )
```

Saves an image using a specified file format (using the extension of the filename)

## See also

spinError

## Parameters

|                  |                                                                                        |
|------------------|----------------------------------------------------------------------------------------|
| <i>hImage</i>    | The image to be saved                                                                  |
| <i>pFilename</i> | The filename to use to save the image (with or without the appropriate file extension) |

## Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

**6.5.1.101 spinImageSaveJpeg()**

```
SPINNAKERC_API spinImageSaveJpeg (
    spinImage hImage,
    const char * pFilename,
    const spinJPEGOption * pOption )
```

Saves an image as a JPEG image.

## See also

`spinError`

## Parameters

|                  |                                                                                                  |
|------------------|--------------------------------------------------------------------------------------------------|
| <i>hImage</i>    | The image to be saved                                                                            |
| <i>pFilename</i> | The filename to use to save the image (with or without the appropriate file extension)           |
| <i>pOption</i>   | The image options related to saving as JPEG; includes quality and whether to save as progressive |

## Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

**6.5.1.102 spinImageSaveJpg2()**

```
SPINNAKERC_API spinImageSaveJpg2 (
    spinImage hImage,
    const char * pFilename,
    const spinJPG2Option * pOption )
```

Saves an image as a JPEG 2000 image.

## See also

`spinError`

## Parameters

|                  |                                                                                        |
|------------------|----------------------------------------------------------------------------------------|
| <i>hImage</i>    | The image to be saved                                                                  |
| <i>pFilename</i> | The filename to use to save the image (with or without the appropriate file extension) |
| <i>pOption</i>   | The image options related to saving as JPEG 2000; includes quality                     |

## Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.103 spinImageSavePgm()**

```
SPINNAKERC_API spinImageSavePgm (  
    spinImage hImage,  
    const char * pFilename,  
    const spinPGMOption * pOption )
```

Saves an image as an PGM image.

## See also

spinError

## Parameters

|                  |                                                                                        |
|------------------|----------------------------------------------------------------------------------------|
| <i>hImage</i>    | The image to be saved                                                                  |
| <i>pFilename</i> | The filename to use to save the image (with or without the appropriate file extension) |
| <i>pOption</i>   | The image options related to saving as PGM; includes whether to save as binary         |

## Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.104 spinImageSavePng()**

```
SPINNAKERC_API spinImageSavePng (  
    spinImage hImage,  
    const char * pFilename,  
    const spinPNGOption * pOption )
```

Saves an image as a PNG image.

## See also

spinError

## Parameters

|                  |                                                                                                          |
|------------------|----------------------------------------------------------------------------------------------------------|
| <i>hImage</i>    | The image to be saved                                                                                    |
| <i>pFilename</i> | The filename to use to save the image (with or without the appropriate file extension)                   |
| <i>pOption</i>   | The image options related to saving as PNG; includes compression level and whether to save as interlaced |

## Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

**6.5.1.105 spinImageSavePpm()**

```
SPINNAKERC_API spinImageSavePpm (  
    spinImage hImage,  
    const char * pFilename,  
    const spinPPMOption * pOption )
```

Saves an image as a PPM image.

## See also

`spinError`

## Parameters

|                  |                                                                                        |
|------------------|----------------------------------------------------------------------------------------|
| <i>hImage</i>    | The image to be saved                                                                  |
| <i>pFilename</i> | The filename to use to save the image (with or without the appropriate file extension) |
| <i>pOption</i>   | The image options related to saving as PPM; includes whether to save as binary         |

## Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

**6.5.1.106 spinImageSaveTiff()**

```
SPINNAKERC_API spinImageSaveTiff (  
    spinImage hImage,  
    const char * pFilename,  
    const spinTIFFOption * pOption )
```

Saves an image as a TIFF image.

## See also

`spinError`

## Parameters

|                  |                                                                                        |
|------------------|----------------------------------------------------------------------------------------|
| <i>hImage</i>    | The image to be saved                                                                  |
| <i>pFilename</i> | The filename to use to save the image (with or without the appropriate file extension) |
| <i>pOption</i>   | The image options related to saving as TIFF; includes compression method               |

## Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.107 spinImageSetDefaultColorProcessing()**

```
SPINNAKERC_API spinImageSetDefaultColorProcessing (
    spinColorProcessingAlgorithm algorithm )
```

Sets the default color processing algorithm of all images (if not otherwise set)

## See also

spinError

## Parameters

|                  |                                                |
|------------------|------------------------------------------------|
| <i>algorithm</i> | The color processing algorithm used by default |
|------------------|------------------------------------------------|

## Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.108 spinImageStatisticsCreate()**

```
SPINNAKERC_API spinImageStatisticsCreate (
    spinImageStatistics * phStatistics )
```

Creates an image statistics context.

## Parameters

|                     |                                                                                 |
|---------------------|---------------------------------------------------------------------------------|
| <i>phStatistics</i> | The statistics handle pointer in which the image statistics context is returned |
|---------------------|---------------------------------------------------------------------------------|

## Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

#### 6.5.1.109 spinImageStatisticsDestroy()

```
SPINNAKERC_API spinImageStatisticsDestroy (  
    spinImageStatistics hStatistics )
```

Destroys an image statistics context.

##### See also

spinError

##### Parameters

|                    |                                         |
|--------------------|-----------------------------------------|
| <i>hStatistics</i> | The image statistics context to destroy |
|--------------------|-----------------------------------------|

##### Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

#### 6.5.1.110 spinImageStatisticsDisableAll()

```
SPINNAKERC_API spinImageStatisticsDisableAll (  
    spinImageStatistics hStatistics )
```

Disables all channels of an image statistics context.

##### See also

spinError

##### Parameters

|                    |                                                      |
|--------------------|------------------------------------------------------|
| <i>hStatistics</i> | The image statistics context to disable all channels |
|--------------------|------------------------------------------------------|

##### Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

#### 6.5.1.111 spinImageStatisticsEnableAll()

```
SPINNAKERC_API spinImageStatisticsEnableAll (  
    spinImageStatistics hStatistics )
```

Enables all channels of an image statistics context.



**See also**

spinError

**Parameters**

|                    |                                                     |
|--------------------|-----------------------------------------------------|
| <i>hStatistics</i> | The image statistics context to enable all channels |
|--------------------|-----------------------------------------------------|

**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.112 spinImageStatisticsEnableGreyOnly()**

```
SPINNAKERC_API spinImageStatisticsEnableGreyOnly (  
    spinImageStatistics hStatistics )
```

Disables all channels of an image statistics context except grey-scale.

**See also**

spinError

**Parameters**

|                    |                                                  |
|--------------------|--------------------------------------------------|
| <i>hStatistics</i> | The image statistics context to enable only grey |
|--------------------|--------------------------------------------------|

**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.113 spinImageStatisticsEnableHslOnly()**

```
SPINNAKERC_API spinImageStatisticsEnableHslOnly (  
    spinImageStatistics hStatistics )
```

Disables all channels of an image statistics context except hue, saturation, and lightness.

**See also**

spinError

## Parameters

|                    |                                                 |
|--------------------|-------------------------------------------------|
| <i>hStatistics</i> | The image statistics context to enable only HSL |
|--------------------|-------------------------------------------------|

## Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

**6.5.1.114 spinImageStatisticsEnableRgbOnly()**

```
SPINNAKERC_API spinImageStatisticsEnableRgbOnly (
    spinImageStatistics hStatistics )
```

Disables all channels of an image statistics context except red, blue, and green.

## See also

`spinError`

## Parameters

|                    |                                                 |
|--------------------|-------------------------------------------------|
| <i>hStatistics</i> | The image statistics context to enable only RGB |
|--------------------|-------------------------------------------------|

## Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

**6.5.1.115 spinImageStatisticsGetAll()**

```
SPINNAKERC_API spinImageStatisticsGetAll (
    spinImageStatistics hStatistics,
    spinStatisticsChannel channel,
    unsigned int * pRangeMin,
    unsigned int * pRangeMax,
    unsigned int * pPixelValueMin,
    unsigned int * pPixelValueMax,
    unsigned int * pNumPixelValues,
    float * pPixelValueMean,
    int ** ppHistogram )
```

Retrieves all available information of an image statistics channel.

## See also

`spinError`

## Parameters

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| <i>hStatistics</i>     | The image statistics context of the channel                                            |
| <i>channel</i>         | The channel of the information to retrieve                                             |
| <i>pRangeMin</i>       | The unsigned integer pointer in which the minimum value of the range is returned       |
| <i>pRangeMax</i>       | The unsigned integer pointer in which the maximum value of the range is returned       |
| <i>pPixelValueMin</i>  | The unsigned integer pointer in which the minimum pixel value of the range is returned |
| <i>pPixelValueMax</i>  | The unsigned integer pointer in which the maximum pixel value of the range is returned |
| <i>pNumPixelValues</i> | The unsigned integer pointer in which the number of pixel values is returned           |
| <i>pPixelValueMean</i> | The float pointer in which the mean pixel value is returned                            |
| <i>ppiHistogram</i>    | The pointer to the pointer in which the histogram data is returned                     |

## Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

## 6.5.1.116 spinImageStatisticsGetChannelStatus()

```
SPINNAKERC_API spinImageStatisticsGetChannelStatus (
    spinImageStatistics hStatistics,
    spinStatisticsChannel channel,
    bool8_t * pbEnabled )
```

Checks whether an image statistics context is enabled.

## See also

spinError

## Parameters

|                    |                                                                     |
|--------------------|---------------------------------------------------------------------|
| <i>hStatistics</i> | The image statistics context of the channel                         |
| <i>channel</i>     | The channel to check                                                |
| <i>pbEnabled</i>   | The boolean pointer to return whether or not the channel is enabled |

## Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

## 6.5.1.117 spinImageStatisticsGetHistogram()

```
SPINNAKERC_API spinImageStatisticsGetHistogram (
    spinImageStatistics hStatistics,
    spinStatisticsChannel channel,
    int ** ppHistogram )
```

Retrieves a histogram of an image statistics channel.

See also

`spinError`

Parameters

|                    |                                                                            |
|--------------------|----------------------------------------------------------------------------|
| <i>hStatistics</i> | The image statistics context of the channel                                |
| <i>channel</i>     | The channel of the histogram to be returned                                |
| <i>pHistogram</i>  | The pointer to the integer pointer in which the histogram data is returned |

Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

#### 6.5.1.118 `spinImageStatisticsGetMean()`

```
SPINNAKERC_API spinImageStatisticsGetMean (
    spinImageStatistics hStatistics,
    spinStatisticsChannel channel,
    float * pMean )
```

Retrieves the mean of pixel values of an image statistics channel.

See also

`spinError`

Parameters

|                    |                                                             |
|--------------------|-------------------------------------------------------------|
| <i>hStatistics</i> | The image statistics context of the channel                 |
| <i>channel</i>     | The channel of the mean pixel value to be retrieved         |
| <i>pMean</i>       | The float pointer in which the mean pixel value is returned |

Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

#### 6.5.1.119 `spinImageStatisticsGetNumPixelValues()`

```
SPINNAKERC_API spinImageStatisticsGetNumPixelValues (
    spinImageStatistics hStatistics,
    spinStatisticsChannel channel,
    unsigned int * pNumValues )
```

Retrieves the number of pixel values of an image statistics channel.

## See also

spinError

## Parameters

|                    |                                                                              |
|--------------------|------------------------------------------------------------------------------|
| <i>hStatistics</i> | The image statistics context of the channel                                  |
| <i>channel</i>     | The channel where the pixel values to be counted are                         |
| <i>iNumValues</i>  | The unsigned integer pointer in which the number of pixel values is returned |

## Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.120 spinImageStatisticsGetPixelValueRange()**

```
SPINNAKERC_API spinImageStatisticsGetPixelValueRange (
    spinImageStatistics hStatistics,
    spinStatisticsChannel channel,
    unsigned int * pMin,
    unsigned int * pMax )
```

Retrieves the pixel value range of an image statistics channel.

## See also

spinError

## Parameters

|                    |                                                                                              |
|--------------------|----------------------------------------------------------------------------------------------|
| <i>hStatistics</i> | The image statistics context of the channel                                                  |
| <i>channel</i>     | The channel of the pixel value range to retrieve                                             |
| <i>pMin</i>        | The unsigned integer pointer in which the minimum value of the pixel value range is returned |
| <i>pMax</i>        | The unsigned integer pointer in which the maximum value of the pixel value range is returned |

## Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.121 spinImageStatisticsGetRange()**

```
SPINNAKERC_API spinImageStatisticsGetRange (
    spinImageStatistics hStatistics,
    spinStatisticsChannel channel,
    unsigned int * pMin,
    unsigned int * pMax )
```

Retrieves the range of an image statistics channel.

See also

`spinError`

Parameters

|                    |                                                                                  |
|--------------------|----------------------------------------------------------------------------------|
| <i>hStatistics</i> | The image statistics context of the channel                                      |
| <i>channel</i>     | The channel of the range to retrieve                                             |
| <i>pMin</i>        | The unsigned integer pointer in which the minimum value of the range is returned |
| <i>pMax</i>        | The unsigned integer pointer in which the maximum value of the range is returned |

Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

#### 6.5.1.122 `spinImageStatisticsSetChannelStatus()`

```
SPINNAKERC_API spinImageStatisticsSetChannelStatus (
    spinImageStatistics hStatistics,
    spinStatisticsChannel channel,
    bool8_t bEnable )
```

Sets the status of an image statistics channel.

See also

`spinError`

Parameters

|                    |                                                        |
|--------------------|--------------------------------------------------------|
| <i>hStatistics</i> | The image statistics context of the channel            |
| <i>channel</i>     | The channel to enable/disable                          |
| <i>bEnable</i>     | The boolean value to set; true enables, false disables |

Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

#### 6.5.1.123 `spinInterfaceEventHandlerCreate()`

```
SPINNAKERC_API spinInterfaceEventHandlerCreate (
    spinInterfaceEventHandler * phInterfaceEventHandler,
    spinArrivalEventFunction pArrivalFunction,
    spinRemovalEventFunction pRemovalFunction,
    void * pUserData )
```

Creates an interface event handler (both device arrival and device removal)

## See also

spinError

## Parameters

|                                |                                                                                                                                                                |
|--------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>phInterfaceEventHandler</i> | The interface event handler pointer in which the interface event context is created                                                                            |
| <i>pArrivalFunction</i>        | The function to be called at arrival event occurrences; signature to match:<br>void(<em>spinArrivalEventFunction)(void pUserData)                              |
| <i>hRemovalFunction</i>        | The function to be called at removal event occurrences; signature to match:<br>void(<em>spinRemovalEventFunction)(uint64_t deviceSerialNumber, void pUserData) |
| <i>pUserData</i>               | Properties that can be passed into the event function                                                                                                          |

## Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.124 spinInterfaceEventHandlerDestroy()**

```
SPINNAKERC_API spinInterfaceEventHandlerDestroy (
    spinInterfaceEventHandler hInterfaceEventHandler )
```

Destroys an interface event handler (both device arrival and device removal)

## See also

spinError

## Parameters

|                               |                                        |
|-------------------------------|----------------------------------------|
| <i>hInterfaceEventHandler</i> | The interface event handler to destroy |
|-------------------------------|----------------------------------------|

## Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.125 spinInterfaceGetCameras()**

```
SPINNAKERC_API spinInterfaceGetCameras (
    spinInterface hInterface,
    spinCameraList hCameraList )
```

Retrieves a camera list from an interface; camera lists must be created and destroy.

## See also

[spinCameraListCreateEmpty\(\)](#)  
[spinCameraListDestroy\(\)](#)  
[spinError](#)

## Parameters

|                    |                                                         |
|--------------------|---------------------------------------------------------|
| <i>hInterface</i>  | The interface of the camera list to retrieve            |
| <i>hCameraList</i> | The camera list to house the cameras from the interface |

## Returns

[spinError](#) The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.126 spinInterfaceGetCamerasEx()**

```

SPINNAKERC_API spinInterfaceGetCamerasEx (
    spinInterface hInterface,
    bool8_t bUpdateCameras,
    spinCameraList hCameraList )

```

Retrieves a camera list from an interface; manually set whether to update the cameras; camera lists must be created and destroyed.

## See also

[spinCameraListCreateEmpty\(\)](#)  
[spinCameraListDestroy\(\)](#)  
[spinError](#)

## Parameters

|                       |                                                         |
|-----------------------|---------------------------------------------------------|
| <i>hInterface</i>     | The interface of the camera list to retrieve            |
| <i>bUpdateCameras</i> | The boolean of whether or not to update the cameras     |
| <i>hCameraList</i>    | The camera list to house the cameras from the interface |

## Returns

[spinError](#) The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.127 spinInterfaceGetTLNodeMap()**

```

SPINNAKERC_API spinInterfaceGetTLNodeMap (
    spinInterface hInterface,
    spinNodeMapHandle * phNodeMap )

```



Retrieves the transport layer nodemap from an interface.

See also

`spinError`

Parameters

|                   |                                                                                       |
|-------------------|---------------------------------------------------------------------------------------|
| <i>hInterface</i> | The interface of the nodemap to retrieve                                              |
| <i>phNodeMap</i>  | The nodemap handle pointer in which the transport layer interface nodemap is returned |

Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

#### 6.5.1.128 `spinInterfaceIsInUse()`

```
SPINNAKERC_API spinInterfaceIsInUse (  
    spinInterface hInterface,  
    bool8_t * pbIsInUse )
```

Checks whether an interface is in use.

See also

`spinError`

Parameters

|                   |                                                                      |
|-------------------|----------------------------------------------------------------------|
| <i>hInterface</i> | The interface to check                                               |
| <i>pbIsInUse</i>  | The boolean pointer to return whether or not the interface is in use |

Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

#### 6.5.1.129 `spinInterfaceListClear()`

```
SPINNAKERC_API spinInterfaceListClear (  
    spinInterfaceList hInterfaceList )
```

Clears an interface list.

See also

`spinError`

## Parameters

|                       |                             |
|-----------------------|-----------------------------|
| <i>hInterfaceList</i> | The interface list to clear |
|-----------------------|-----------------------------|

## Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.130 spinInterfaceListCreateEmpty()**

```
SPINNAKERC_API spinInterfaceListCreateEmpty (  
    spinInterfaceList * phInterfaceList )
```

Creates an empty interface list (interface lists created this way must be destroyed)

## See also

spinError

## Parameters

|                        |                                                                                 |
|------------------------|---------------------------------------------------------------------------------|
| <i>phInterfaceList</i> | The interface list handle pointer in which the empty interface list is returned |
|------------------------|---------------------------------------------------------------------------------|

## Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.131 spinInterfaceListDestroy()**

```
SPINNAKERC_API spinInterfaceListDestroy (  
    spinInterfaceList hInterfaceList )
```

Destroys an interface list.

## See also

spinError

## Parameters

|                       |                               |
|-----------------------|-------------------------------|
| <i>hInterfaceList</i> | The interface list to destroy |
|-----------------------|-------------------------------|

**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.132 spinInterfaceListGet()**

```
SPINNAKERC_API spinInterfaceListGet (
    spinInterfaceList hInterfaceList,
    size_t index,
    spinInterface * phInterface )
```

Retrieves an interface from an interface list using an index (interfaces retrieved this way must be released)

**See also**

spinError

**Parameters**

|                       |                                                                 |
|-----------------------|-----------------------------------------------------------------|
| <i>hInterfaceList</i> | The interface list of the interface to be retrieved             |
| <i>index</i>          | The index of the interface                                      |
| <i>phInterface</i>    | The interface handle pointer in which the interface is returned |

**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.133 spinInterfaceListGetSize()**

```
SPINNAKERC_API spinInterfaceListGetSize (
    spinInterfaceList hInterfaceList,
    size_t * pSize )
```

Retrieves the number of interfaces in an interface list.

**See also**

spinError

**Parameters**

|                       |                                                                            |
|-----------------------|----------------------------------------------------------------------------|
| <i>hInterfaceList</i> | The interface list where the interfaces to be counted are                  |
| <i>pSize</i>          | The unsigned integer pointer in which the number of interfaces is returned |

**Returns**

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

**See also**

`spinError`

**6.5.1.134 spinInterfaceRegisterDeviceArrivalEventHandler()**

```
SPINNAKERC_API spinInterfaceRegisterDeviceArrivalEventHandler (
    spinInterface hInterface,
    spinDeviceArrivalEventHandler hDeviceArrivalEventHandler )
```

Registers a device arrival event handler on an interface (event handlers registered in this way must be unregistered)

**See also**

`spinError`

**Parameters**

|                                   |                                                                     |
|-----------------------------------|---------------------------------------------------------------------|
| <i>hInterface</i>                 | The interface on which to register the device arrival event handler |
| <i>hDeviceArrivalEventHandler</i> | The device arrival event handler to register                        |

**Returns**

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

**6.5.1.135 spinInterfaceRegisterDeviceRemovalEventHandler()**

```
SPINNAKERC_API spinInterfaceRegisterDeviceRemovalEventHandler (
    spinInterface hInterface,
    spinDeviceRemovalEventHandler hDeviceRemovalEventHandler )
```

Registers a device removal event handler on an interface (event handlers registered in this way must be unregistered)

**See also**

`spinError`

**Parameters**

|                                   |                                                                     |
|-----------------------------------|---------------------------------------------------------------------|
| <i>hInterface</i>                 | the Interface on which to register the device removal event handler |
| <i>hDeviceRemovalEventHandler</i> | The device removal event handler to register                        |

**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.136 spinInterfaceRegisterInterfaceEventHandler()**

```
SPINNAKERC_API spinInterfaceRegisterInterfaceEventHandler (
    spinInterface hInterface,
    spinInterfaceEventHandler hInterfaceEventHandler )
```

Registers an interface event handler (both device arrival and device removal) on an interface.

**See also**

spinError

**Parameters**

|                               |                                                                |
|-------------------------------|----------------------------------------------------------------|
| <i>hInterface</i>             | The interface on which to register the interface event handler |
| <i>hInterfaceEventHandler</i> | The interface event handler to register                        |

**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.137 spinInterfaceRelease()**

```
SPINNAKERC_API spinInterfaceRelease (
    spinInterface hInterface )
```

Releases an interface.

**See also**

spinError

**Parameters**

|                   |                          |
|-------------------|--------------------------|
| <i>hInterface</i> | The interface to release |
|-------------------|--------------------------|

**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

### 6.5.1.138 spinInterfaceSendActionCommand()

```
SPINNAKERC_API spinInterfaceSendActionCommand (
    spinInterface hInterface,
    size_t iDeviceKey,
    size_t iGroupKey,
    size_t iGroupMask,
    size_t iActionTime,
    size_t * piResultSize,
    actionCommandResult results[] )
```

Broadcast an Action Command to all devices on interface.

See also

spinError

#### Parameters

|                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>iDeviceKey</i>   | The Action Command's device key                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <i>iGroupKey</i>    | The Action Command's group key                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <i>iGroupMask</i>   | The Action Command's group mask                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <i>iActionTime</i>  | (Optional) Time when to assert a future action. Zero means immediate action.                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <i>piResultSize</i> | (Optional) The number of results in the results array. The value passed should be equal to the expected number of devices that acknowledge the command. Returns the number of received results.                                                                                                                                                                                                                                                                                                                                  |
| <i>results</i>      | (Optional) An Array with *piResultSize elements to hold the action command result status. The buffer is filled starting from index 0. If received results are less than expected number of devices that acknowledge the command, remaining results are not changed. If received results are more than expected number of devices that acknowledge the command, extra results are ignored and not appended to array. This parameter is ignored if piResultSize is 0. Thus this parameter can be NULL if pResultSize is 0 or NULL. |

#### Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

### 6.5.1.139 spinInterfaceUnregisterDeviceArrivalEventHandler()

```
SPINNAKERC_API spinInterfaceUnregisterDeviceArrivalEventHandler (
    spinInterface hInterface,
    spinDeviceArrivalEventHandler hDeviceArrivalEventHandler )
```

Unregisters a device arrival event handler from an interface.

See also

spinError

## Parameters

|                                   |                                                                         |
|-----------------------------------|-------------------------------------------------------------------------|
| <i>hInterface</i>                 | The interface from which to unregister the device arrival event handler |
| <i>hDeviceArrivalEventHandler</i> | The device arrival event handler to unregister                          |

## Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.140 spinInterfaceUnregisterDeviceRemovalEventHandler()**

```
SPINNAKERC_API spinInterfaceUnregisterDeviceRemovalEventHandler (
    spinInterface hInterface,
    spinDeviceRemovalEventHandler hDeviceRemovalEventHandler )
```

Unregisters a device removal event handler from an interface.

## See also

spinError

## Parameters

|                                   |                                                                         |
|-----------------------------------|-------------------------------------------------------------------------|
| <i>hInterface</i>                 | The interface from which to unregister the device removal event handler |
| <i>hDeviceRemovalEventHandler</i> | The device removal event handler to unregister                          |

## Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.141 spinInterfaceUnregisterInterfaceEventHandler()**

```
SPINNAKERC_API spinInterfaceUnregisterInterfaceEventHandler (
    spinInterface hInterface,
    spinInterfaceEventHandler hInterfaceEventHandler )
```

Unregisters an interface event handler from an interface.

## See also

spinError

## Parameters

|                               |                                                                    |
|-------------------------------|--------------------------------------------------------------------|
| <i>hInterface</i>             | The interface from which to unregister the interface event handler |
| <i>hInterfaceEventHandler</i> | The interface event handler to unregister                          |

## Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.142 spinInterfaceUpdateCameras()**

```
SPINNAKERC_API spinInterfaceUpdateCameras (
    spinInterface hInterface,
    bool8_t * pbChanged )
```

Checks whether any cameras have been connected or disconnected on an interface.

## See also

spinError

## Parameters

|                   |                                                                       |
|-------------------|-----------------------------------------------------------------------|
| <i>hInterface</i> | The interface of the list of attached cameras to update               |
| <i>pbChanged</i>  | The boolean pointer to return whether or not the cameras have changed |

## Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.143 spinLogDataGetCategoryName()**

```
SPINNAKERC_API spinLogDataGetCategoryName (
    spinLogEventData hLogEventData,
    char * pBuf,
    size_t * pBufLen )
```

Retrieves the category name of a log event.

## See also

spinError



## Parameters

|                      |                                                                                                                     |
|----------------------|---------------------------------------------------------------------------------------------------------------------|
| <i>hLogEventData</i> | The log event data received from the log event                                                                      |
| <i>pBuf</i>          | The c-string character buffer in which the category name of the log event is returned                               |
| <i>pBufLen</i>       | The unsigned integer pointer in which the length of the c-string is returned; the input value is the maximum length |

## Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

**6.5.1.144 spinLogDataGetLogMessage()**

```
SPINNAKERC_API spinLogDataGetLogMessage (
    spinLogEventData hLogEventData,
    char * pBuf,
    size_t * pBufLen )
```

Retrieves the log message of a log event.

## See also

`spinError`

## Parameters

|                      |                                                                                                                     |
|----------------------|---------------------------------------------------------------------------------------------------------------------|
| <i>hLogEventData</i> | The log event data received from the log event                                                                      |
| <i>pBuf</i>          | The c-string character buffer in which the log message of the log event is returned                                 |
| <i>pBufLen</i>       | The unsigned integer pointer in which the length of the c-string is returned; the input value is the maximum length |

## Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

**6.5.1.145 spinLogDataGetNDC()**

```
SPINNAKERC_API spinLogDataGetNDC (
    spinLogEventData hLogEventData,
    char * pBuf,
    size_t * pBufLen )
```

Retrieves the NDC of a log event.

## See also

`spinError`

## Parameters

|                      |                                                                                                                     |
|----------------------|---------------------------------------------------------------------------------------------------------------------|
| <i>hLogEventData</i> | The log event data received from the log event                                                                      |
| <i>pBuf</i>          | The c-string character buffer in which the NDC of the log event is returned                                         |
| <i>pBufLen</i>       | The unsigned integer pointer in which the length of the c-string is returned; the input value is the maximum length |

## Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

**6.5.1.146 spinLogDataGetPriority()**

```
SPINNAKERC_API spinLogDataGetPriority (
    spinLogEventData hLogEventData,
    int64_t * pValue )
```

Retrieves the priority of a log event.

## See also

`spinError`

## Parameters

|                      |                                                             |
|----------------------|-------------------------------------------------------------|
| <i>hLogEventData</i> | The log event data received from the log event              |
| <i>pValue</i>        | The integer pointer in which the priority value is returned |

## Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

**6.5.1.147 spinLogDataGetPriorityName()**

```
SPINNAKERC_API spinLogDataGetPriorityName (
    spinLogEventData hLogEventData,
    char * pBuf,
    size_t * pBufLen )
```

Retrieves the priority name of a log event.

## See also

`spinError`

## Parameters

|                      |                                                                                                                     |
|----------------------|---------------------------------------------------------------------------------------------------------------------|
| <i>hLogEventData</i> | The log event data received from the log event                                                                      |
| <i>pBuf</i>          | The c-string character buffer in which the priority name of the log event is returned                               |
| <i>pBufLen</i>       | The unsigned integer pointer in which the length of the c-string is returned; the input value is the maximum length |

## Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.148 spinLogDataGetThreadName()**

```
SPINNAKERC_API spinLogDataGetThreadName (  
    spinLogEventData hLogEventData,  
    char * pBuf,  
    size_t * pBufLen )
```

Retrieves the thread name of a log event.

## See also

spinError

## Parameters

|                      |                                                                                                                     |
|----------------------|---------------------------------------------------------------------------------------------------------------------|
| <i>hLogEventData</i> | The log event data received from the log event                                                                      |
| <i>pBuf</i>          | The c-string character buffer in which the thread name of the log event is returned                                 |
| <i>pBufLen</i>       | The unsigned integer pointer in which the length of the c-string is returned; the input value is the maximum length |

## Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.149 spinLogDataGetTimestamp()**

```
SPINNAKERC_API spinLogDataGetTimestamp (  
    spinLogEventData hLogEventData,  
    char * pBuf,  
    size_t * pBufLen )
```

Retrieves the timestamp of a log event.

## See also

spinError

## Parameters

|                      |                                                                                                                     |
|----------------------|---------------------------------------------------------------------------------------------------------------------|
| <i>hLogEventData</i> | The log event data received from the log event                                                                      |
| <i>pBuf</i>          | The c-string character buffer in which the timestamp of the log event is returned                                   |
| <i>pBufLen</i>       | The unsigned integer pointer in which the length of the c-string is returned; the input value is the maximum length |

## Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

**6.5.1.150 spinLogEventHandlerCreate()**

```
SPINNAKERC_API spinLogEventHandlerCreate (
    spinLogEventHandler * phLogEventHandler,
    spinLogEventFunction pFunction,
    void * pUserData )
```

Creates a log event handler.

## See also

`spinError`

## Parameters

|                          |                                                                                                                                                                 |
|--------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>phLogEventHandler</i> | The log event handler pointer in which the log event context is created                                                                                         |
| <i>pFunction</i>         | The function to be called at device event occurrences; signature to match:<br>void(<em>spinLogEventFunction)(const spinLogEventData hEventData, void pUserData) |
| <i>pUserData</i>         | Properties that can be passed into the event function                                                                                                           |

## Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

**6.5.1.151 spinLogEventHandlerDestroy()**

```
SPINNAKERC_API spinLogEventHandlerDestroy (
    spinLogEventHandler hLogEventHandler )
```

Destroys a log event handler.

## See also

`spinError`

## Parameters

|                         |                                  |
|-------------------------|----------------------------------|
| <i>hLogEventHandler</i> | The log event handler to destroy |
|-------------------------|----------------------------------|

## Returns

*spinError* The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.152 spinSystemGetCameras()**

```
SPINNAKERC_API spinSystemGetCameras (  
    spinSystem hSystem,  
    spinCameraList hCameraList )
```

Retrieves a list of detected (and enumerable) cameras on the system; camera lists must be created and destroyed.

## See also

[spinCameraListCreateEmpty\(\)](#)

[spinCameraListDestroy\(\)](#)

*spinError*

## Parameters

|                    |                                                      |
|--------------------|------------------------------------------------------|
| <i>hSystem</i>     | The system, from which the camera list is retrieved  |
| <i>hCameraList</i> | The camera list to house the cameras from the system |

## Returns

*spinError* The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.153 spinSystemGetCamerasEx()**

```
SPINNAKERC_API spinSystemGetCamerasEx (  
    spinSystem hSystem,  
    bool8_t bUpdateInterfaces,  
    bool8_t bUpdateCameras,  
    spinCameraList hCameraList )
```

Retrieves a list of detected (and enumerable) cameras on the system; manually set whether to update the current interface and camera lists; camera lists must be created and destroyed.

## See also

[spinCameraListCreateEmpty\(\)](#)

[spinCameraListDestroy\(\)](#)

*spinError*

## Parameters

|                          |                                                      |
|--------------------------|------------------------------------------------------|
| <i>hSystem</i>           | The system, from which the camera list is retrieved  |
| <i>bUpdateInterfaces</i> | The boolean of whether to update the interface list  |
| <i>bUpdateCameras</i>    | The boolean of whether to update the camera list     |
| <i>hCameraList</i>       | The camera list to house the cameras from the system |

## Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.154 spinSystemGetInstance()**

```
SPINNAKERC_API spinSystemGetInstance (
    spinSystem * phSystem )
```

Retrieves an instance of the system object; the system is a singleton, so there will only ever be one instance; system instance must be destroyed by calling spinSystemReleaseInstance.

## See also

[spinSystemReleaseInstance](#)

spinError

## Parameters

|                 |                                                                    |
|-----------------|--------------------------------------------------------------------|
| <i>phSystem</i> | The system handle pointer in which the system instance is returned |
|-----------------|--------------------------------------------------------------------|

## Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.155 spinSystemGetInterfaces()**

```
SPINNAKERC_API spinSystemGetInterfaces (
    spinSystem hSystem,
    spinInterfaceList hInterfaceList )
```

Retrieves a list of detected (and enumerable) interfaces on the system; interface lists must be created and destroyed.

## See also

[spinInterfaceListCreateEmpty\(\)](#)

[spinInterfaceListDestroy\(\)](#)

spinError

## Parameters

|                       |                                                            |
|-----------------------|------------------------------------------------------------|
| <i>hSystem</i>        | The system, from which the interface list is retrieved     |
| <i>hInterfaceList</i> | The interface list to house the interfaces from the system |

## Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.156 spinSystemGetLibraryVersion()**

```
SPINNAKERC_API spinSystemGetLibraryVersion (
    spinSystem hSystem,
    spinLibraryVersion * hLibraryVersion )
```

Get current library version of Spinnaker.

## Returns

A struct containing the current version of Spinnaker(major, minor, type, build).

**6.5.1.157 spinSystemGetLoggingLevel()**

```
SPINNAKERC_API spinSystemGetLoggingLevel (
    spinSystem hSystem,
    spinnakerLogLevel * pLogLevel )
```

Retrieves the logging level for all logging events on the system.

## See also

spinError

## Parameters

|                 |                                                                               |
|-----------------|-------------------------------------------------------------------------------|
| <i>hSystem</i>  | The system, from which the logging level is retrieved                         |
| <i>logLevel</i> | The logging level enum pointer in which the current logging level is returned |

## Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

#### 6.5.1.158 spinSystemGetTLNodeMap()

```
SPINNAKERC_API spinSystemGetTLNodeMap (
    spinSystem hSystem,
    spinNodeMapHandle * phNodeMap )
```

Retrieves the transport layer nodemap from the system.

See also

spinError

Parameters

|                  |                                                                                     |
|------------------|-------------------------------------------------------------------------------------|
| <i>hSystem</i>   | The system handle.                                                                  |
| <i>phNodeMap</i> | The nodemap handle pointer in which the transport layer system nodemap is returned. |

Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

#### 6.5.1.159 spinSystemIsInUse()

```
SPINNAKERC_API spinSystemIsInUse (
    spinSystem hSystem,
    bool8_t * pbIsInUse )
```

Checks whether a system is currently in use.

See also

spinError

Parameters

|                  |                                                                      |
|------------------|----------------------------------------------------------------------|
| <i>hSystem</i>   | The system to check                                                  |
| <i>pbIsInUse</i> | The boolean pointer to return whether the system is currently in use |

Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

#### 6.5.1.160 spinSystemRegisterDeviceArrivalEventHandler()

```
SPINNAKERC_API spinSystemRegisterDeviceArrivalEventHandler (
```



```
spinSystem hSystem,  
spinDeviceArrivalEventHandler hDeviceArrivalEventHandler )
```

Registers a device arrival event handler to every interface on the system (event handlers registered this way must be unregistered)

See also

spinError

Parameters

|                                   |                                                                     |
|-----------------------------------|---------------------------------------------------------------------|
| <i>hSystem</i>                    | The system, on which the device arrival event handler is registered |
| <i>hDeviceArrivalEventHandler</i> | The device arrival event handler to register on the system          |

Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

#### 6.5.1.161 spinSystemRegisterDeviceRemovalEventHandler()

```
SPINNAKERC_API spinSystemRegisterDeviceRemovalEventHandler (  
    spinSystem hSystem,  
    spinDeviceRemovalEventHandler hDeviceRemovalEventHandler )
```

Registers a device removal event handler to the system to every interface on the system (event handlers registered this way must be unregistered)

See also

spinError

Parameters

|                                   |                                                                     |
|-----------------------------------|---------------------------------------------------------------------|
| <i>hSystem</i>                    | The system, on which the device removal event handler is registered |
| <i>hDeviceRemovalEventHandler</i> | The device removal event handler to register on the system          |

Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

#### 6.5.1.162 spinSystemRegisterInterfaceEventHandler()

```
SPINNAKERC_API spinSystemRegisterInterfaceEventHandler (  
    spinSystem hSystem,  
    spinInterfaceEventHandler hInterfaceEventHandler )
```

Registers an interface event handler (device arrival and device removal) to every interface on the system (interface events registered this way must be unregistered) If new interfaces are detected by the system after `spinSystemRegisterInterfaceEventHandler()` is called, those interfaces will be automatically registered with this event.

See also

`spinError`

Parameters

|                               |                                                                                           |
|-------------------------------|-------------------------------------------------------------------------------------------|
| <i>hSystem</i>                | The system, on which the interface event handler is registered                            |
| <i>hInterfaceEventHandler</i> | The interface event handler (device arrival and device removal) to register on the system |

Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

#### 6.5.1.163 `spinSystemRegisterLogEventHandler()`

```
SPINNAKERC_API spinSystemRegisterLogEventHandler (
    spinSystem hSystem,
    spinLogEventHandler hLogEventHandler )
```

Registers a logging event handler to the system (event handlers registered in this way must be unregistered)

See also

`spinError`

Parameters

|                         |                                                              |
|-------------------------|--------------------------------------------------------------|
| <i>hSystem</i>          | The system, on which the logging event handler is registered |
| <i>hLogEventHandler</i> | The logging event handler to register on the system          |

Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

#### 6.5.1.164 `spinSystemReleaseInstance()`

```
SPINNAKERC_API spinSystemReleaseInstance (
    spinSystem hSystem )
```

Releases the system; make sure handle is cleaned up properly by setting it to NULL after system is released; the handle can only be used again after calling `spinSystemGetInstance`.

See also

[spinSystemGetInstance](#)

[spinError](#)

Parameters

|                |                   |
|----------------|-------------------|
| <i>hSystem</i> | The system handle |
|----------------|-------------------|

Returns

[spinError](#) The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

#### 6.5.1.165 spinSystemSendActionCommand()

```
SPINNAKERC_API spinSystemSendActionCommand (
    spinSystem hSystem,
    size_t iDeviceKey,
    size_t iGroupKey,
    size_t iGroupMask,
    size_t iActionTime,
    size_t * piResultSize,
    actionCommandResult results[] )
```

Broadcast an Action Command to all devices on system.

See also

[spinError](#)

Parameters

|                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>hSystem</i>      | The system on which to send the action command to all devices.                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <i>iDeviceKey</i>   | The Action Command's device key                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <i>iGroupKey</i>    | The Action Command's group key                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <i>iGroupMask</i>   | The Action Command's group mask                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <i>iActionTime</i>  | (Optional) Time when to assert a future action. Zero means immediate action.                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <i>piResultSize</i> | (Optional) The number of results in the results array. The value passed should be equal to the expected number of devices that acknowledge the command. Returns the number of received results.                                                                                                                                                                                                                                                                                                                                  |
| <i>results</i>      | (Optional) An Array with *piResultSize elements to hold the action command result status. The buffer is filled starting from index 0. If received results are less than expected number of devices that acknowledge the command, remaining results are not changed. If received results are more than expected number of devices that acknowledge the command, extra results are ignored and not appended to array. This parameter is ignored if piResultSize is 0. Thus this parameter can be NULL if pResultSize is 0 or NULL. |

**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.166 spinSystemSetLoggingLevel()**

```
SPINNAKERC_API spinSystemSetLoggingLevel (
    spinSystem hSystem,
    spinnakerLogLevel logLevel )
```

Sets the logging level for all logging events on the system.

**See also**

spinError

**Parameters**

|                 |                                               |
|-----------------|-----------------------------------------------|
| <i>hSystem</i>  | The system, on which the logging level is set |
| <i>logLevel</i> | The logging level to set                      |

**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.5.1.167 spinSystemUnregisterAllLogEventHandlers()**

```
SPINNAKERC_API spinSystemUnregisterAllLogEventHandlers (
    spinSystem hSystem )
```

Unregisters all logging event handlers from the system.

**See also**

spinError

**Parameters**

|                |                                                                    |
|----------------|--------------------------------------------------------------------|
| <i>hSystem</i> | The system, from which all logging event handlers are unregistered |
|----------------|--------------------------------------------------------------------|

**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

#### 6.5.1.168 spinSystemUnregisterDeviceArrivalEventHandler()

```
SPINNAKERC_API spinSystemUnregisterDeviceArrivalEventHandler (
    spinSystem hSystem,
    spinDeviceArrivalEventHandler hDeviceArrivalEventHandler )
```

Unregisters a device arrival event handler from the system.

##### See also

spinError

##### Parameters

|                                   |                                                                         |
|-----------------------------------|-------------------------------------------------------------------------|
| <i>hSystem</i>                    | The system, from which the device arrival event handler is unregistered |
| <i>hDeviceArrivalEventHandler</i> | The device arrival event handler to unregister from the system          |

##### Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

#### 6.5.1.169 spinSystemUnregisterDeviceRemovalEventHandler()

```
SPINNAKERC_API spinSystemUnregisterDeviceRemovalEventHandler (
    spinSystem hSystem,
    spinDeviceRemovalEventHandler hDeviceRemovalEventHandler )
```

Unregisters a device removal event handler from the system.

##### See also

spinError

##### Parameters

|                                   |                                                                         |
|-----------------------------------|-------------------------------------------------------------------------|
| <i>hSystem</i>                    | The system, from which the device removal event handler is unregistered |
| <i>hDeviceRemovalEventHandler</i> | The device removal event handler to unregister from the system          |

##### Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

#### 6.5.1.170 spinSystemUnregisterInterfaceEventHandler()

```
SPINNAKERC_API spinSystemUnregisterInterfaceEventHandler (
    spinSystem hSystem,
    spinInterfaceEventHandler hInterfaceEventHandler )
```

Unregisters an interface event handler from the system.

##### See also

spinError

##### Parameters

|                               |                                                                                               |
|-------------------------------|-----------------------------------------------------------------------------------------------|
| <i>hSystem</i>                | The system, from which the interface event handler is unregistered                            |
| <i>hInterfaceEventHandler</i> | The interface event handler (device arrival and device removal) to unregister from the system |

##### Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

#### 6.5.1.171 spinSystemUnregisterLogEventHandler()

```
SPINNAKERC_API spinSystemUnregisterLogEventHandler (
    spinSystem hSystem,
    spinLogEventHandler hLogEventHandler )
```

Unregisters a selected logging event handler from the system.

##### See also

spinError

##### Parameters

|                         |                                                                  |
|-------------------------|------------------------------------------------------------------|
| <i>hSystem</i>          | The system, from which the logging event handler is unregistered |
| <i>hLogEventHandler</i> | The logging event handler to unregister from the system          |

##### Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

### 6.5.1.172 spinSystemUpdateCameras()

```
SPINNAKERC_API spinSystemUpdateCameras (
    spinSystem hSystem,
    bool8_t * pbChanged )
```

Updates the list of cameras on the system, informing whether there has been any changes.

See also

spinError

#### Parameters

|                  |                                                                                               |
|------------------|-----------------------------------------------------------------------------------------------|
| <i>hSystem</i>   | The system, on which the list of attached cameras is updated                                  |
| <i>pbChanged</i> | The boolean pointer to return whether cameras have arrived on or been removed from the system |

#### Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

### 6.5.1.173 spinSystemUpdateCamerasEx()

```
SPINNAKERC_API spinSystemUpdateCamerasEx (
    spinSystem hSystem,
    bool8_t bUpdateInterfaces,
    bool8_t * pbChanged )
```

Updates the list of cameras on the system, informing whether there has been any changes; manually set whether to update the current interface lists.

See also

spinError

#### Parameters

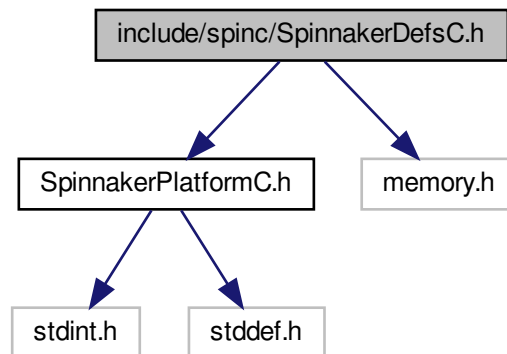
|                          |                                                                                            |
|--------------------------|--------------------------------------------------------------------------------------------|
| <i>hSystem</i>           | The system, on which the list of attached cameras is updated                               |
| <i>bUpdateInterfaces</i> | The boolean of whether to update the interface list                                        |
| <i>pbChanged</i>         | The boolean pointer to return whether cameras have arrived or been removed from the system |

## Returns

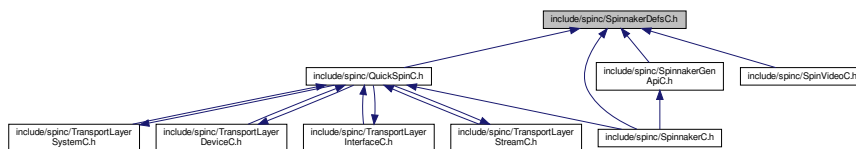
spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

## 6.6 include/spinc/SpinnakerDefsC.h File Reference

Include dependency graph for SpinnakerDefsC.h:



This graph shows which files directly or indirectly include this file:



## Data Structures

- [struct \\_spinPNGOption](#)  
Options for saving PNG images.
- [struct \\_spinPPMOption](#)  
Options for saving PPM images.
- [struct \\_spinPGMOption](#)  
Options for saving PGM images.
- [struct \\_spinTIFFOption](#)  
Options for saving TIFF images.
- [struct \\_spinJPEGOption](#)  
Options for saving JPEG images.
- [struct \\_spinJPG2Option](#)  
Options for saving JPEG 2000 images.



- struct [\\_spinBMPOption](#)  
*Options for saving BMP images.*
- struct [\\_spinMJPEGOption](#)  
*Options for saving MJPG videos.*
- struct [\\_spinH264Option](#)  
*Options for saving H264 videos.*
- struct [\\_spinAVIOption](#)  
*Options for saving uncompressed videos.*
- struct [\\_spinLibraryVersion](#)  
*Provides easier access to the current version of Spinnaker.*
- struct [\\_actionCommandResult](#)  
*Action Command Result.*

## Typedefs

- typedef uint8\_t [bool8\\_t](#)
- typedef void \* [spinSystem](#)  
*Handle for system functionality.*
- typedef void \* [spinInterfaceList](#)  
*Handle for interface list functionality.*
- typedef void \* [spinInterface](#)  
*Handle for interface functionality.*
- typedef void \* [spinCameraList](#)  
*Handle for interface functionality.*
- typedef void \* [spinCamera](#)  
*Handle for camera functionality.*
- typedef void \* [spinImage](#)  
*Handle for image functionality.*
- typedef void \* [spinImageStatistics](#)  
*Handle for image statistics functionality.*
- typedef void \* [spinDeviceEventHandler](#)  
*Handle for device event handler functionality.*
- typedef void \* [spinImageEventHandler](#)  
*Handle for image event handler functionality.*
- typedef void \* [spinDeviceArrivalEventHandler](#)  
*Handle for arrival event handler functionality.*
- typedef void \* [spinDeviceRemovalEventHandler](#)  
*Handle for removal event handler functionality.*
- typedef void \* [spinInterfaceEventHandler](#)  
*Handle for interface event handler functionality.*
- typedef void \* [spinLogEventHandler](#)  
*Handle for logging event handler functionality.*
- typedef void \* [spinLogEventData](#)  
*Handle for logging event data functionality.*
- typedef void \* [spinDeviceEventData](#)  
*Handle for device event data functionality.*
- typedef void \* [spinVideo](#)  
*Handle for video recording functionality.*
- typedef void(\* [spinDeviceEventFunction](#)) (const [spinDeviceEventData](#) hEventData, const char \*pEventName, void \*pUserData)

*Function signatures are used to create and trigger callbacks and events.*

- typedef void(\* [spinImageEventFunction](#)) (const [spinImage](#) hImage, void \*pUserData)
- typedef void(\* [spinArrivalEventFunction](#)) (uint64\_t deviceSerialNumber, void \*pUserData)
- typedef void(\* [spinRemovalEventFunction](#)) (uint64\_t deviceSerialNumber, void \*pUserData)
- typedef void(\* [spinLogEventFunction](#)) (const [spinLogEventData](#) hEventData, void \*pUserData)

## Enumerations

- enum [\\_spinError](#) {  
[SPINNAKER\\_ERR\\_SUCCESS](#) = 0,  
[SPINNAKER\\_ERR\\_ERROR](#) = -1001,  
[SPINNAKER\\_ERR\\_NOT\\_INITIALIZED](#) = -1002,  
[SPINNAKER\\_ERR\\_NOT\\_IMPLEMENTED](#) = -1003,  
[SPINNAKER\\_ERR\\_RESOURCE\\_IN\\_USE](#) = -1004,  
[SPINNAKER\\_ERR\\_ACCESS\\_DENIED](#) = -1005,  
[SPINNAKER\\_ERR\\_INVALID\\_HANDLE](#) = -1006,  
[SPINNAKER\\_ERR\\_INVALID\\_ID](#) = -1007,  
[SPINNAKER\\_ERR\\_NO\\_DATA](#) = -1008,  
[SPINNAKER\\_ERR\\_INVALID\\_PARAMETER](#) = -1009,  
[SPINNAKER\\_ERR\\_IO](#) = -1010,  
[SPINNAKER\\_ERR\\_TIMEOUT](#) = -1011,  
[SPINNAKER\\_ERR\\_ABORT](#) = -1012,  
[SPINNAKER\\_ERR\\_INVALID\\_BUFFER](#) = -1013,  
[SPINNAKER\\_ERR\\_NOT\\_AVAILABLE](#) = -1014,  
[SPINNAKER\\_ERR\\_INVALID\\_ADDRESS](#) = -1015,  
[SPINNAKER\\_ERR\\_BUFFER\\_TOO\\_SMALL](#) = -1016,  
[SPINNAKER\\_ERR\\_INVALID\\_INDEX](#) = -1017,  
[SPINNAKER\\_ERR\\_PARSING\\_CHUNK\\_DATA](#) = -1018,  
[SPINNAKER\\_ERR\\_INVALID\\_VALUE](#) = -1019,  
[SPINNAKER\\_ERR\\_RESOURCE\\_EXHAUSTED](#) = -1020,  
[SPINNAKER\\_ERR\\_OUT\\_OF\\_MEMORY](#) = -1021,  
[SPINNAKER\\_ERR\\_BUSY](#) = -1022,  
[GENICAM\\_ERR\\_INVALID\\_ARGUMENT](#) = -2001,  
[GENICAM\\_ERR\\_OUT\\_OF\\_RANGE](#) = -2002,  
[GENICAM\\_ERR\\_PROPERTY](#) = -2003,  
[GENICAM\\_ERR\\_RUN\\_TIME](#) = -2004,  
[GENICAM\\_ERR\\_LOGICAL](#) = -2005,  
[GENICAM\\_ERR\\_ACCESS](#) = -2006,  
[GENICAM\\_ERR\\_TIMEOUT](#) = -2007,  
[GENICAM\\_ERR\\_DYNAMIC\\_CAST](#) = -2008,  
[GENICAM\\_ERR\\_GENERIC](#) = -2009,  
[GENICAM\\_ERR\\_BAD\\_ALLOCATION](#) = -2010,  
[SPINNAKER\\_ERR\\_IM\\_CONVERT](#) = -3001,  
[SPINNAKER\\_ERR\\_IM\\_COPY](#) = -3002,  
[SPINNAKER\\_ERR\\_IM\\_MALLOC](#) = -3003,  
[SPINNAKER\\_ERR\\_IM\\_NOT\\_SUPPORTED](#) = -3004,  
[SPINNAKER\\_ERR\\_IM\\_HISTOGRAM\\_RANGE](#) = -3005,  
[SPINNAKER\\_ERR\\_IM\\_HISTOGRAM\\_MEAN](#) = -3006,  
[SPINNAKER\\_ERR\\_IM\\_MIN\\_MAX](#) = -3007,  
[SPINNAKER\\_ERR\\_IM\\_COLOR\\_CONVERSION](#) = -3008,  
[SPINNAKER\\_ERR\\_CUSTOM\\_ID](#) = -10000 }

*The error codes used in Spinnaker C.*

- enum [\\_spinColorProcessingAlgorithm](#) {  
[DEFAULT](#),  
[NO\\_COLOR\\_PROCESSING](#),  
[NEAREST\\_NEIGHBOR](#),

```

NEAREST_NEIGHBOR_AVG,
BILINEAR,
EDGE_SENSING,
HQ_LINEAR,
IPP,
DIRECTIONAL_FILTER,
RIGOROUS,
WEIGHTED_DIRECTIONAL_FILTER }

```

*Color processing algorithms.*

- enum `_spinStatisticsChannel` {  
`GREY`,  
`RED`,  
`GREEN`,  
`BLUE`,  
`HUE`,  
`SATURATION`,  
`LIGHTNESS`,  
`NUM_STATISTICS_CHANNELS` }

*Channels that allow statistics to be calculated.*

- enum `_spinImageFileFormat` {  
`FROM_FILE_EXT` = -1,  
`PGM`,  
`PPM`,  
`BMP`,  
`JPEG`,  
`JPEG2000`,  
`TIFF`,  
`PNG`,  
`RAW`,  
`IMAGE_FILE_FORMAT_FORCE_32BITS` = 0x7FFFFFFF }

*File formats to be used for saving images to disk.*

- enum `_spinPixelFormatNamespaceID` {  
`SPINNAKER_PIXELFORMAT_NAMESPACE_UNKNOWN` = 0,  
`SPINNAKER_PIXELFORMAT_NAMESPACE_GEV` = 1,  
`SPINNAKER_PIXELFORMAT_NAMESPACE_IIDC` = 2,  
`SPINNAKER_PIXELFORMAT_NAMESPACE_PFNC_16BIT` = 3,  
`SPINNAKER_PIXELFORMAT_NAMESPACE_PFNC_32BIT` = 4,  
`SPINNAKER_PIXELFORMAT_NAMESPACE_CUSTOM_ID` = 1000 }

*This enum represents the namespace in which the TL specific pixel format resides.*

- enum `_spinImageStatus` {  
`IMAGE_UNKNOWN_ERROR` = -1,  
`IMAGE_NO_ERROR` = 0,  
`IMAGE_CRC_CHECK_FAILED` = 1,  
`IMAGE_DATA_OVERFLOW` = 2,  
`IMAGE_MISSING_PACKETS` = 3,  
`IMAGE_LEADER_BUFFER_SIZE_INCONSISTENT` = 4,  
`IMAGE_TRAILER_BUFFER_SIZE_INCONSISTENT` = 5,  
`IMAGE_PACKETID_INCONSISTENT` = 6,  
`IMAGE_MISSING_LEADER` = 7,  
`IMAGE_MISSING_TRAILER` = 8,  
`IMAGE_DATA_INCOMPLETE` = 9,  
`IMAGE_INFO_INCONSISTENT` = 10,  
`IMAGE_CHUNK_DATA_INVALID` = 11,  
`IMAGE_NO_SYSTEM_RESOURCES` = 12 }

*Status of images returned from `spinImageGetStatus()` call.*

- enum `_spinLogLevel` {  
`LOG_LEVEL_OFF` = -1,

```

LOG_LEVEL_FATAL = 0,
LOG_LEVEL_ALERT = 100,
LOG_LEVEL_CRIT = 200,
LOG_LEVEL_ERROR = 300,
LOG_LEVEL_WARN = 400,
LOG_LEVEL_NOTICE = 500,
LOG_LEVEL_INFO = 600,
LOG_LEVEL_DEBUG = 700,
LOG_LEVEL_NOTSET = 800 }

```

*log levels*

- enum `_spinPayloadTypeInfoIds` {  
`PAYLOAD_TYPE_UNKNOWN` = 0,  
`PAYLOAD_TYPE_IMAGE` = 1,  
`PAYLOAD_TYPE_RAW_DATA` = 2,  
`PAYLOAD_TYPE_FILE` = 3,  
`PAYLOAD_TYPE_CHUNK_DATA` = 4,  
`PAYLOAD_TYPE_JPEG` = 5,  
`PAYLOAD_TYPE_JPEG2000` = 6,  
`PAYLOAD_TYPE_H264` = 7,  
`PAYLOAD_TYPE_CHUNK_ONLY` = 8,  
`PAYLOAD_TYPE_DEVICE_SPECIFIC` = 9,  
`PAYLOAD_TYPE_MULTI_PART` = 10,  
`PAYLOAD_TYPE_CUSTOM_ID` = 1000,  
`PAYLOAD_TYPE_EXTENDED_CHUNK` = 1001 }
- enum `CompressionMethod` {  
`NONE` = 1,  
`PACKBITS`,  
`DEFLATE`,  
`ADOBE_DEFLATE`,  
`CCITTFAX3`,  
`CCITTFAX4`,  
`LZW`,  
`JPG` }

*Compression method used in saving TIFF images in the spinTIFFOption struct.*

- enum `_actionCommandStatus` {  
`ACTION_COMMAND_STATUS_OK` = 0,  
`ACTION_COMMAND_STATUS_NO_REF_TIME` = 0x8013,  
`ACTION_COMMAND_STATUS_OVERFLOW` = 0x8015,  
`ACTION_COMMAND_STATUS_ACTION_LATE` = 0x8016,  
`ACTION_COMMAND_STATUS_ERROR` = 0x8FFF }

*Possible Status Codes Returned from Action Command.*

## Variables

- static const `bool8_t False` = 0
- static const `bool8_t True` = 1

### 6.6.1 Typedef Documentation

#### 6.6.1.1 bool8\_t

```
typedef uint8_t bool8_t
```

### 6.6.1.2 spinArrivalEventFunction

```
typedef void(* spinArrivalEventFunction) (uint64_t deviceSerialNumber, void *pUserData)
```

### 6.6.1.3 spinCamera

```
typedef void* spinCamera
```

Handle for camera functionality.

Created by calling [spinCameraListGet\(\)](#), which requires a call to [spinCameraRelease\(\)](#) to release.

### 6.6.1.4 spinCameraList

```
typedef void* spinCameraList
```

Handle for interface functionality.

Created by calling [spinSystemGetCameras\(\)](#) or [spinInterfaceGetCameras\(\)](#), which require a call to [spinCameraListClear\(\)](#) to clear, or [spinCameraListCreateEmpty\(\)](#), which requires a call to [spinCameraListDestroy\(\)](#) to destroy.

### 6.6.1.5 spinDeviceArrivalEventHandler

```
typedef void* spinDeviceArrivalEventHandler
```

Handle for arrival event handler functionality.

Created by calling [spinArrivalEventCreate\(\)](#), which requires a call to [spinDeviceArrivalEventHandlerDestroy\(\)](#) to destroy.

### 6.6.1.6 spinDeviceEventData

```
typedef void* spinDeviceEventData
```

Handle for device event data functionality.

Received in device event function. No need to release, clear, or destroy.

### 6.6.1.7 spinDeviceEventFunction

```
typedef void(* spinDeviceEventFunction) (const spinDeviceEventData hEventData, const char *pEventName, void *pUserData)
```

Function signatures are used to create and trigger callbacks and events.

#### 6.6.1.8 spinDeviceEventHandler

```
typedef void* spinDeviceEventHandler
```

Handle for device event handler functionality.

Created by calling [spinDeviceEventHandlerCreate\(\)](#), which requires a call to [spinDeviceEventHandlerDestroy\(\)](#) to destroy.

#### 6.6.1.9 spinDeviceRemovalEventHandler

```
typedef void* spinDeviceRemovalEventHandler
```

Handle for removal event handler functionality.

Created by calling [spinDeviceRemovalEventHandlerCreate\(\)](#), which requires a call to [spinDeviceRemovalEventHandlerDestroy\(\)](#) to destroy.

#### 6.6.1.10 spinImage

```
typedef void* spinImage
```

Handle for image functionality.

Created by calling [spinCameraGetNextImage\(\)](#) or [spinCameraGetNextImageEx\(\)](#), which require a call to [spinImageRelease\(\)](#) to remove from buffer, or [spinImageCreateEmpty\(\)](#), [spinImageCreateEx\(\)](#), or [spinImageCreate\(\)](#), which require a call to [spinImageDestroy\(\)](#) to destroy.

#### 6.6.1.11 spinImageEventFunction

```
typedef void(* spinImageEventFunction) (const spinImage hImage, void *pUserData)
```

#### 6.6.1.12 spinImageEventHandler

```
typedef void* spinImageEventHandler
```

Handle for image event handler functionality.

Created by calling [spinImageEventHandlerCreate\(\)](#), which requires a call to [spinImageEventHandlerDestroy\(\)](#) to destroy.

#### 6.6.1.13 spinImageStatistics

```
typedef void* spinImageStatistics
```

Handle for image statistics functionality.

Created by calling [spinImageStatisticsCreate\(\)](#), which requires a call to [spinImageStatisticsDestroy\(\)](#) to destroy.

#### 6.6.1.14 spinInterface

```
typedef void* spinInterface
```

Handle for interface functionality.

Created by calling [spinInterfaceListGet\(\)](#), which requires a call to [spinInterfaceRelease\(\)](#) to release.

#### 6.6.1.15 spinInterfaceEventHandler

```
typedef void* spinInterfaceEventHandler
```

Handle for interface event handler functionality.

Created by calling [spinInterfaceEventHandlerCreate\(\)](#), which requires a call to [spinInterfaceEventHandlerDestroy\(\)](#) to destroy.

#### 6.6.1.16 spinInterfaceList

```
typedef void* spinInterfaceList
```

Handle for interface list functionality.

Created by calling [spinSystemGetInterfaces\(\)](#), which requires a call to [spinInterfaceListClear\(\)](#) to clear, or [spinInterfaceListCreateEmpty\(\)](#), which requires a call to [spinInterfaceListDestroy\(\)](#) to destroy.

#### 6.6.1.17 spinLogEventData

```
typedef void* spinLogEventData
```

Handle for logging event data functionality.

Received in log event function. No need to release, clear, or destroy.

#### 6.6.1.18 spinLogEventFunction

```
typedef void(* spinLogEventFunction) (const spinLogEventData hEventData, void *pUserData)
```

#### 6.6.1.19 spinLogEventHandler

```
typedef void* spinLogEventHandler
```

Handle for logging event handler functionality.

Created by calling [spinLogEventHandlerCreate\(\)](#), which requires a call to [spinLogEventHandlerDestroy\(\)](#) to destroy.

### 6.6.1.20 spinRemovalEventFunction

```
typedef void(* spinRemovalEventFunction) (uint64_t deviceSerialNumber, void *pUserData)
```

### 6.6.1.21 spinSystem

```
typedef void* spinSystem
```

Handle for system functionality.

Created by calling [spinSystemGetInstance\(\)](#), which requires a call to [spinSystemReleaseInstance\(\)](#) to release.

### 6.6.1.22 spinVideo

```
typedef void* spinVideo
```

Handle for video recording functionality.

Created by calling [spinVideoOpenUncompressed\(\)](#), [spinVideoOpenMJPEG\(\)](#), and [spinVideoOpenH264\(\)](#), which require a call to [spinVideoClose\(\)](#) to destroy.

## 6.6.2 Enumeration Type Documentation

### 6.6.2.1 \_actionCommandStatus

```
enum _actionCommandStatus
```

Possible Status Codes Returned from Action Command.

Enumerator

|                                   |                                      |
|-----------------------------------|--------------------------------------|
| ACTION_COMMAND_STATUS_OK          | The device acknowledged the command. |
| ACTION_COMMAND_STATUS_NO_REF_TIME |                                      |
| ACTION_COMMAND_STATUS_OVERFLOW    |                                      |
| ACTION_COMMAND_STATUS_ACTION_LATE |                                      |
| ACTION_COMMAND_STATUS_ERROR       |                                      |

### 6.6.2.2 \_spinColorProcessingAlgorithm

```
enum _spinColorProcessingAlgorithm
```



Color processing algorithms.

Please refer to our knowledge base at article at <https://www.flir.com/support-center/iis/machine-vision/k> for complete details for each algorithm.

#### Enumerator

|                             |                                                                                                                        |
|-----------------------------|------------------------------------------------------------------------------------------------------------------------|
| DEFAULT                     | Default method.                                                                                                        |
| NO_COLOR_PROCESSING         | No color processing.                                                                                                   |
| NEAREST_NEIGHBOR            | Fastest but lowest quality. Equivalent to FLYCAPTURE_NEAREST_NEIGHBOR_FAST in FlyCapture.                              |
| NEAREST_NEIGHBOR_AVG        | Nearest Neighbor with averaged green pixels. Higher quality but slower compared to nearest neighbor without averaging. |
| BILINEAR                    | Weighted average of surrounding 4 pixels in a 2x2 neighborhood.                                                        |
| EDGE_SENSING                | Weights surrounding pixels based on localized edge orientation.                                                        |
| HQ_LINEAR                   | Well-balanced speed and quality.                                                                                       |
| IPP                         | Multi-threaded with similar results to edge sensing.                                                                   |
| DIRECTIONAL_FILTER          | Best quality but much faster than rigorous.                                                                            |
| RIGOROUS                    | Slowest but produces good results.                                                                                     |
| WEIGHTED_DIRECTIONAL_FILTER | Weighted pixel average from different directions.                                                                      |

#### 6.6.2.3 \_spinError

enum `_spinError`

The error codes used in Spinnaker C.

These codes are returned from every function in Spinnaker C. The error codes in the range of -2000 to -2999 are reserved for GenICam related errors. The error codes in the range of -3000 to -3999 are reserved for image processing related errors.

#### Enumerator

|                                 |                                                                                       |
|---------------------------------|---------------------------------------------------------------------------------------|
| SPINNAKER_ERR_SUCCESS           | An error code of 0 means that the function has run without error.                     |
| SPINNAKER_ERR_ERROR             | The error codes in the range of -1000 to -1999 are reserved for Spinnaker exceptions. |
| SPINNAKER_ERR_NOT_INITIALIZED   |                                                                                       |
| SPINNAKER_ERR_NOT_IMPLEMENTED   |                                                                                       |
| SPINNAKER_ERR_RESOURCE_IN_USE   |                                                                                       |
| SPINNAKER_ERR_ACCESS_DENIED     |                                                                                       |
| SPINNAKER_ERR_INVALID_HANDLE    |                                                                                       |
| SPINNAKER_ERR_INVALID_ID        |                                                                                       |
| SPINNAKER_ERR_NO_DATA           |                                                                                       |
| SPINNAKER_ERR_INVALID_PARAMETER |                                                                                       |
| SPINNAKER_ERR_IO                |                                                                                       |
| SPINNAKER_ERR_TIMEOUT           |                                                                                       |
| SPINNAKER_ERR_ABORT             |                                                                                       |
| SPINNAKER_ERR_INVALID_BUFFER    |                                                                                       |

## Enumerator

|                                   |                                                                                                  |
|-----------------------------------|--------------------------------------------------------------------------------------------------|
| SPINNAKER_ERR_NOT_AVAILABLE       |                                                                                                  |
| SPINNAKER_ERR_INVALID_ADDRESS     |                                                                                                  |
| SPINNAKER_ERR_BUFFER_TOO_SMALL    |                                                                                                  |
| SPINNAKER_ERR_INVALID_INDEX       |                                                                                                  |
| SPINNAKER_ERR_PARSING_CHUNK_DATA  |                                                                                                  |
| SPINNAKER_ERR_INVALID_VALUE       |                                                                                                  |
| SPINNAKER_ERR_RESOURCE_EXHAUSTED  |                                                                                                  |
| SPINNAKER_ERR_OUT_OF_MEMORY       |                                                                                                  |
| SPINNAKER_ERR_BUSY                |                                                                                                  |
| GENICAM_ERR_INVALID_ARGUMENT      | The error codes in the range of -2000 to -2999 are reserved for Gen API related errors.          |
| GENICAM_ERR_OUT_OF_RANGE          |                                                                                                  |
| GENICAM_ERR_PROPERTY              |                                                                                                  |
| GENICAM_ERR_RUN_TIME              |                                                                                                  |
| GENICAM_ERR_LOGICAL               |                                                                                                  |
| GENICAM_ERR_ACCESS                |                                                                                                  |
| GENICAM_ERR_TIMEOUT               |                                                                                                  |
| GENICAM_ERR_DYNAMIC_CAST          |                                                                                                  |
| GENICAM_ERR_GENERIC               |                                                                                                  |
| GENICAM_ERR_BAD_ALLOCATION        |                                                                                                  |
| SPINNAKER_ERR_IM_CONVERT          | The error codes in the range of -3000 to -3999 are reserved for image processing related errors. |
| SPINNAKER_ERR_IM_COPY             |                                                                                                  |
| SPINNAKER_ERR_IM_MALLOC           |                                                                                                  |
| SPINNAKER_ERR_IM_NOT_SUPPORTED    |                                                                                                  |
| SPINNAKER_ERR_IM_HISTOGRAM_RANGE  |                                                                                                  |
| SPINNAKER_ERR_IM_HISTOGRAM_MEAN   |                                                                                                  |
| SPINNAKER_ERR_IM_MIN_MAX          |                                                                                                  |
| SPINNAKER_ERR_IM_COLOR_CONVERSION |                                                                                                  |
| SPINNAKER_ERR_CUSTOM_ID           | Error codes less than -10000 are reserved for user-defined custom errors.                        |

## 6.6.2.4 \_spinImageFileFormat

```
enum _spinImageFileFormat
```

File formats to be used for saving images to disk.

## Enumerator

|               |                                            |
|---------------|--------------------------------------------|
| FROM_FILE_EXT | Determine file format from file extension. |
| PGM           | Portable gray map.                         |
| PPM           | Portable pixmap.                           |
| BMP           | Bitmap.                                    |
| JPEG          | JPEG.                                      |
| JPEG2000      | JPEG 2000.                                 |
| TIFF          | Tagged image file format.                  |

## Enumerator

|                                |                            |
|--------------------------------|----------------------------|
| PNG                            | Portable network graphics. |
| RAW                            | Raw data.                  |
| IMAGE_FILE_FORMAT_FORCE_32BITS |                            |

## 6.6.2.5 \_spinImageStatus

enum `_spinImageStatus`

Status of images returned from `spinImageGetStatus()` call.

## Enumerator

|                                        |                                                                                                                                                                                                                                                                                         |
|----------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| IMAGE_UNKNOWN_ERROR                    | Image has an unknown error.                                                                                                                                                                                                                                                             |
| IMAGE_NO_ERROR                         | Image is returned from <code>GetNextImage()</code> call without any errors.                                                                                                                                                                                                             |
| IMAGE_CRC_CHECK_FAILED                 | Image failed CRC check.                                                                                                                                                                                                                                                                 |
| IMAGE_DATA_OVERFLOW                    | Received more data than the size of the image.                                                                                                                                                                                                                                          |
| IMAGE_MISSING_PACKETS                  | Image has missing packets. Potential fixes include enabling jumbo packets and adjusting packet size/delay. For more information see <a href="https://www.flir.com/support-center/iis/machine-vision/application">https://www.flir.com/support-center/iis/machine-vision/application</a> |
| IMAGE_LEADER_BUFFER_SIZE_INCONSISTENT  | Image leader is incomplete. Could be caused by missing packet(s). See link above.                                                                                                                                                                                                       |
| IMAGE_TRAILER_BUFFER_SIZE_INCONSISTENT | Image trailer is incomplete. Could be caused by missing packet(s). See link above.                                                                                                                                                                                                      |
| IMAGE_PACKETID_INCONSISTENT            | Image has an inconsistent packet id. Could be caused by missing packet(s). See link above.                                                                                                                                                                                              |
| IMAGE_MISSING_LEADER                   | Image leader is missing. Could be caused by missing packet(s). See link above.                                                                                                                                                                                                          |
| IMAGE_MISSING_TRAILER                  | Image trailer is missing. Could be caused by missing packet(s). See link above.                                                                                                                                                                                                         |
| IMAGE_DATA_INCOMPLETE                  | Image data is incomplete. Could be caused by missing packet(s). See link above.                                                                                                                                                                                                         |
| IMAGE_INFO_INCONSISTENT                | Image info is corrupted. Could be caused by missing packet(s). See link above.                                                                                                                                                                                                          |
| IMAGE_CHUNK_DATA_INVALID               | Image chunk data is invalid.                                                                                                                                                                                                                                                            |
| IMAGE_NO_SYSTEM_RESOURCES              | Image cannot be processed due to lack of system resources.                                                                                                                                                                                                                              |

## 6.6.2.6 \_spinLogLevel

enum `_spinLogLevel`

log levels

## Enumerator

|                  |  |
|------------------|--|
| LOG_LEVEL_OFF    |  |
| LOG_LEVEL_FATAL  |  |
| LOG_LEVEL_ALERT  |  |
| LOG_LEVEL_CRIT   |  |
| LOG_LEVEL_ERROR  |  |
| LOG_LEVEL_WARN   |  |
| LOG_LEVEL_NOTICE |  |
| LOG_LEVEL_INFO   |  |
| LOG_LEVEL_DEBUG  |  |
| LOG_LEVEL_NOTSET |  |

6.6.2.7 `_spinPayloadTypeInfoIDs`

```
enum _spinPayloadTypeInfoIDs
```

## Enumerator

|                              |  |
|------------------------------|--|
| PAYLOAD_TYPE_UNKNOWN         |  |
| PAYLOAD_TYPE_IMAGE           |  |
| PAYLOAD_TYPE_RAW_DATA        |  |
| PAYLOAD_TYPE_FILE            |  |
| PAYLOAD_TYPE_CHUNK_DATA      |  |
| PAYLOAD_TYPE_JPEG            |  |
| PAYLOAD_TYPE_JPEG2000        |  |
| PAYLOAD_TYPE_H264            |  |
| PAYLOAD_TYPE_CHUNK_ONLY      |  |
| PAYLOAD_TYPE_DEVICE_SPECIFIC |  |
| PAYLOAD_TYPE_MULTI_PART      |  |
| PAYLOAD_TYPE_CUSTOM_ID       |  |
| PAYLOAD_TYPE_EXTENDED_CHUNK  |  |

6.6.2.8 `_spinPixelFormatNamespaceID`

```
enum _spinPixelFormatNamespaceID
```

This enum represents the namespace in which the TL specific pixel format resides.

This enum is returned from a captured image when calling [spinImageGetTLPixelFormatNamespace\(\)](#). It can be used to interpret the raw pixel format returned from [spinImageGetTLPixelFormat\(\)](#).

## See also

[spinImageGetTLPixelFormat\(\)](#)

[spinImageGetTLPixelFormatNamespace\(\)](#)

## Enumerator

|                                            |  |
|--------------------------------------------|--|
| SPINNAKER_PIXELFORMAT_NAMESPACE_UNKNOWN    |  |
| SPINNAKER_PIXELFORMAT_NAMESPACE_GEV        |  |
| SPINNAKER_PIXELFORMAT_NAMESPACE_IIDC       |  |
| SPINNAKER_PIXELFORMAT_NAMESPACE_PFNC_16BIT |  |
| SPINNAKER_PIXELFORMAT_NAMESPACE_PFNC_32BIT |  |
| SPINNAKER_PIXELFORMAT_NAMESPACE_CUSTOM_ID  |  |

6.6.2.9 `_spinStatisticsChannel`

enum `_spinStatisticsChannel`

Channels that allow statistics to be calculated.

## Enumerator

|                         |  |
|-------------------------|--|
| GREY                    |  |
| RED                     |  |
| GREEN                   |  |
| BLUE                    |  |
| HUE                     |  |
| SATURATION              |  |
| LIGHTNESS               |  |
| NUM_STATISTICS_CHANNELS |  |

6.6.2.10 `CompressionMethod`

enum `CompressionMethod`

Compression method used in saving TIFF images in the `spinTIFFOption` struct.

## Enumerator

|               |  |
|---------------|--|
| NONE          |  |
| PACKBITS      |  |
| DEFLATE       |  |
| ADOBE_DEFLATE |  |
| CCITTFAX3     |  |
| CCITTFAX4     |  |
| LZW           |  |
| JPG           |  |

## 6.6.3 Variable Documentation

### 6.6.3.1 False

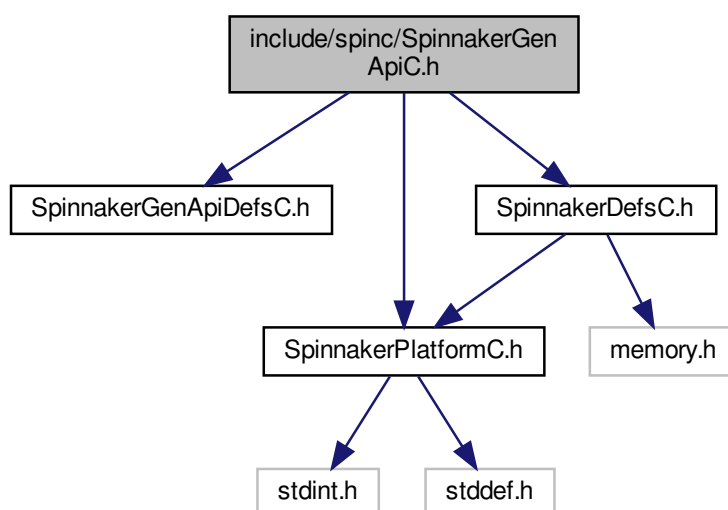
```
const bool8_t False = 0 [static]
```

### 6.6.3.2 True

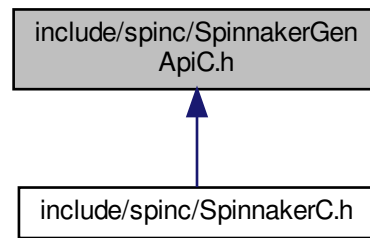
```
const bool8_t True = 1 [static]
```

## 6.7 include/spinc/SpinnakerGenApiC.h File Reference

Include dependency graph for SpinnakerGenApiC.h:



This graph shows which files directly or indirectly include this file:



## Functions

- [SPINNAKERC\\_API spinNodeMapGetNode](#) ([spinNodeMapHandle](#) hNodeMap, [const char](#) \*pName, [spinNodeHandle](#) \*phNode)  
*Retrieves a node from the nodemap by name.*
- [SPINNAKERC\\_API spinNodeMapGetNumNodes](#) ([spinNodeMapHandle](#) hNodeMap, [size\\_t](#) \*pValue)  
*Gets the number of nodes in the map.*
- [SPINNAKERC\\_API spinNodeMapGetNodeByIndex](#) ([spinNodeMapHandle](#) hNodeMap, [size\\_t](#) index, [spinNodeHandle](#) \*phNode)  
*Retrieves a node from the nodemap by index.*
- [SPINNAKERC\\_API spinNodeMapPoll](#) ([spinNodeMapHandle](#) hNodeMap, [int64\\_t](#) timestamp)  
*Fires nodes which have a polling time.*
- [SPINNAKERC\\_API spinNodesImplemented](#) ([spinNodeHandle](#) hNode, [bool8\\_t](#) \*pbResult)  
*Checks whether a node is implemented.*
- [SPINNAKERC\\_API spinNodesReadable](#) ([spinNodeHandle](#) hNode, [bool8\\_t](#) \*pbResult)  
*Checks whether a node is readable.*
- [SPINNAKERC\\_API spinNodesWritable](#) ([spinNodeHandle](#) hNode, [bool8\\_t](#) \*pbResult)  
*Checks whether a node is writable.*
- [SPINNAKERC\\_API spinNodesAvailable](#) ([spinNodeHandle](#) hNode, [bool8\\_t](#) \*pbResult)  
*Checks whether a node is available.*
- [SPINNAKERC\\_API spinNodesEqual](#) ([spinNodeHandle](#) hNodeFirst, [spinNodeHandle](#) hNodeSecond, [bool8\\_t](#) \*pbResult)  
*Checks whether two nodes are equal.*
- [SPINNAKERC\\_API spinNodeGetAccessMode](#) ([spinNodeHandle](#) hNode, [spinAccessMode](#) \*pAccessMode)  
*Retrieves the access mode of a node (as an enum, spinAccessMode)*
- [SPINNAKERC\\_API spinNodeGetName](#) ([spinNodeHandle](#) hNode, [char](#) \*pBuf, [size\\_t](#) \*pBufLen)  
*Retrieves the name of a node (no whitespace)*
- [SPINNAKERC\\_API spinNodeGetNameSpace](#) ([spinNodeHandle](#) hNode, [spinNameSpace](#) \*pNamespace)  
*Retrieve the namespace of a node (as an enum, spinNameSpace)*
- [SPINNAKERC\\_API spinNodeGetVisibility](#) ([spinNodeHandle](#) hNode, [spinVisibility](#) \*pVisibility)  
*Retrieves the recommended visibility of a node (as an enum, spinVisibility)*
- [SPINNAKERC\\_API spinNodeInvalidateNode](#) ([spinNodeHandle](#) hNode)  
*Invalidates a node in case its values may have changed, rendering it no longer valid.*
- [SPINNAKERC\\_API spinNodeGetCachingMode](#) ([spinNodeHandle](#) hNode, [spinCachingMode](#) \*pCachingMode)↵

- Retrieves the caching mode of a node (as an enum, spinCachingMode)*

  - [SPINNAKERC\\_API spinNodeGetToolTip](#) (spinNodeHandle hNode, char \*pBuf, size\_t \*pBufLen)

*Retrieves a short description of a node.*
- [SPINNAKERC\\_API spinNodeGetDescription](#) (spinNodeHandle hNode, char \*pBuf, size\_t \*pBufLen)

*Retrieves a longer description of a node.*
- [SPINNAKERC\\_API spinNodeGetDisplayName](#) (spinNodeHandle hNode, char \*pBuf, size\_t \*pBufLen)

*Retrieves the display name of a node (whitespace possible)*
- [SPINNAKERC\\_API spinNodeGetType](#) (spinNodeHandle hNode, spinNodeType \*pType)

*Retrieves the type of a node (as an enum, spinNodeType)*
- [SPINNAKERC\\_API spinNodeGetPollingTime](#) (spinNodeHandle hNode, int64\_t \*pPollingTime)

*Retrieve the polling time of a node.*
- [SPINNAKERC\\_API spinNodeRegisterCallback](#) (spinNodeHandle hNode, spinNodeCallbackFunction pCb↔  
Function, spinNodeCallbackHandle \*phCb)

*Registers a callback to a node.*
- [SPINNAKERC\\_API spinNodeDeregisterCallback](#) (spinNodeHandle hNode, spinNodeCallbackHandle hCb)

*Unregisters a callback from a node.*
- [SPINNAKERC\\_API spinNodeGetImposedAccessMode](#) (spinNodeHandle hNode, spinAccessMode imposedAccessMode)

*Retrieves the imposed access mode of a node.*
- [SPINNAKERC\\_API spinNodeGetImposedVisibility](#) (spinNodeHandle hNode, spinVisibility imposedVisibility)

*Retrieves the imposed visibility of a node.*
- [SPINNAKERC\\_API spinNodeToString](#) (spinNodeHandle hNode, char \*pBuf, size\_t \*pBufLen)

*Retrieves the value of any node type as a c-string.*
- [SPINNAKERC\\_API spinNodeToStringEx](#) (spinNodeHandle hNode, bool8\_t bVerify, char \*pBuf, size\_t \*p↔  
BufLen)

*Retrieves the value of any node type as a c-string; manually set whether to verify the node.*
- [SPINNAKERC\\_API spinNodeFromString](#) (spinNodeHandle hNode, const char \*pBuf)

*Sets the value of any node type from a c-string; it is important to ensure that the value of the c-string is appropriate to the node type.*
- [SPINNAKERC\\_API spinNodeFromStringEx](#) (spinNodeHandle hNode, bool8\_t bVerify, const char \*pBuf)

*Sets the value of any node type from a c-string; manually set whether to verify the node; ensure the value of the c-string is appropriate to the node type.*
- [SPINNAKERC\\_API spinStringSetValue](#) (spinNodeHandle hNode, const char \*pBuf)

*Sets the value of a string node.*
- [SPINNAKERC\\_API spinStringSetValueEx](#) (spinNodeHandle hNode, bool8\_t bVerify, const char \*pBuf)

*Sets the value of a string node; manually set whether to verify the node.*
- [SPINNAKERC\\_API spinStringGetValue](#) (spinNodeHandle hNode, char \*pBuf, size\_t \*pBufLen)

*Retrieves the value of a string node as a c-string.*
- [SPINNAKERC\\_API spinStringGetValueEx](#) (spinNodeHandle hNode, bool8\_t bVerify, char \*pBuf, size\_t \*p↔  
BufLen)

*Retrieves the value of a string node as a cstring; manually set whether to verify the node.*
- [SPINNAKERC\\_API spinStringGetMaxLength](#) (spinNodeHandle hNode, int64\_t \*pValue)

*Retrieves the maximum length of the c-string to be returned.*
- [SPINNAKERC\\_API spinIntegerSetValue](#) (spinNodeHandle hNode, int64\_t value)

*Sets the value of an integer node.*
- [SPINNAKERC\\_API spinIntegerSetValueEx](#) (spinNodeHandle hNode, bool8\_t bVerify, int64\_t value)

*Sets the value of an integer node; manually set whether to verify the node.*
- [SPINNAKERC\\_API spinIntegerGetValue](#) (spinNodeHandle hNode, int64\_t \*pValue)

*Retrieves the value of an integer node.*
- [SPINNAKERC\\_API spinIntegerGetValueEx](#) (spinNodeHandle hNode, bool8\_t bVerify, int64\_t \*pValue)

*Retrieves the value of an integer node; manually set whether to verify the node.*
- [SPINNAKERC\\_API spinIntegerGetMin](#) (spinNodeHandle hNode, int64\_t \*pValue)



- Retrieves the minimum value of an integer node; all potential values must be greater than or equal to the minimum.*

  - [SPINNAKERC\\_API spinIntegerGetMax](#) ([spinNodeHandle](#) hNode, [int64\\_t](#) \*pValue)

*Retrieves the maximum value of an integer node; all potential values must be lesser than or equal to the maximum.*

  - [SPINNAKERC\\_API spinIntegerGetInc](#) ([spinNodeHandle](#) hNode, [int64\\_t](#) \*pValue)

*Retrieves the increment of an integer node; all possible values must be divisible by the increment.*

  - [SPINNAKERC\\_API spinIntegerGetRepresentation](#) ([spinNodeHandle](#) hNode, [spinRepresentation](#) \*pValue)

*Retrieves the numerical representation of the value of a node; i.e.*

  - [SPINNAKERC\\_API spinFloatSetValue](#) ([spinNodeHandle](#) hNode, [double](#) value)

*Sets the value of a float node.*

  - [SPINNAKERC\\_API spinFloatSetValueEx](#) ([spinNodeHandle](#) hNode, [bool8\\_t](#) bVerify, [double](#) value)

*Sets the value of a float node; manually set whether to verify the node.*

  - [SPINNAKERC\\_API spinFloatGetValue](#) ([spinNodeHandle](#) hNode, [double](#) \*pValue)

*Retrieves the value of a float node.*

  - [SPINNAKERC\\_API spinFloatGetValueEx](#) ([spinNodeHandle](#) hNode, [bool8\\_t](#) bVerify, [double](#) \*pValue)

*Retrieves the value of a float node; manually set whether to verify the node.*

  - [SPINNAKERC\\_API spinFloatGetMin](#) ([spinNodeHandle](#) hNode, [double](#) \*pValue)

*Retrieves the minimum value of a float node; all potential values must be greater than or equal to the minimum.*

  - [SPINNAKERC\\_API spinFloatGetMax](#) ([spinNodeHandle](#) hNode, [double](#) \*pValue)

*Retrieves the maximum value of a float node; all potential values must be lesser than or equal to the maximum.*

  - [SPINNAKERC\\_API spinFloatGetRepresentation](#) ([spinNodeHandle](#) hNode, [spinRepresentation](#) \*pValue)

*Retrieves the numerical representation of the value of a node; i.e.*

  - [SPINNAKERC\\_API spinFloatGetUnit](#) ([spinNodeHandle](#) hNode, [char](#) \*pBuf, [size\\_t](#) \*pBufLen)

*Retrieves the units of the float node value.*

  - [SPINNAKERC\\_API spinEnumerationGetNumEntries](#) ([spinNodeHandle](#) hNode, [size\\_t](#) \*pValue)

*Retrieves the number of entries of an enum node.*

  - [SPINNAKERC\\_API spinEnumerationGetEntryByIndex](#) ([spinNodeHandle](#) hNode, [size\\_t](#) index, [spinNodeHandle](#) \*phEntry)

*Retrieves an entry node from an enum node using an index.*

  - [SPINNAKERC\\_API spinEnumerationGetEntryByName](#) ([spinNodeHandle](#) hNode, [const char](#) \*pName, [spinNodeHandle](#) \*phEntry)

*Retrieves an entry node from an enum node using the entry's symbolic.*

  - [SPINNAKERC\\_API spinEnumerationGetCurrentEntry](#) ([spinNodeHandle](#) hNode, [spinNodeHandle](#) \*phEntry)

*Retrieves the currently selected entry node from an enum node.*

  - [SPINNAKERC\\_API spinEnumerationSetIntValue](#) ([spinNodeHandle](#) hNode, [int64\\_t](#) value)

*Sets a new entry using its integer value retrieved from a call to [spinEnumerationEntryGetIntValue\(\)](#); note that enumeration entry int and enum values are different - int values defined on camera, enum values found in [SpinnakerDefsC.h](#).*

  - [SPINNAKERC\\_API spinEnumerationSetEnumValue](#) ([spinNodeHandle](#) hNode, [size\\_t](#) value)

*Sets a new entry using its enum; note that enumeration entry int and enum values are different - int values defined on camera, enum values found in [SpinnakerDefsC.h](#).*

  - [SPINNAKERC\\_API spinEnumerationEntryGetIntValue](#) ([spinNodeHandle](#) hNode, [int64\\_t](#) \*pValue)

*Retrieves the integer value of an entry node; note that enumeration entry int and enum values are different - int values defined on camera, enum values found in [SpinnakerDefsC.h](#).*

  - [SPINNAKERC\\_API spinEnumerationEntryGetEnumValue](#) ([spinNodeHandle](#) hNode, [size\\_t](#) \*pValue)

*Retrieves the enum value (as an integer) of an entry node; note that enumeration entry int and enum values are different - int values defined on camera, enum values found in [SpinnakerDefsC.h](#).*

  - [SPINNAKERC\\_API spinEnumerationEntryGetSymbolic](#) ([spinNodeHandle](#) hNode, [char](#) \*pBuf, [size\\_t](#) \*pBufLen)

*Retrieves the symbolic of an entry node as a c-string.*

  - [SPINNAKERC\\_API spinBooleanSetValue](#) ([spinNodeHandle](#) hNode, [bool8\\_t](#) value)

*Sets the value of a boolean node; boolean values are represented by 'True' (which equals '0') and 'False' (which equals '1')*

  - [SPINNAKERC\\_API spinBooleanGetValue](#) ([spinNodeHandle](#) hNode, [bool8\\_t](#) \*pValue)

Retrieves the value of a boolean node; boolean values are represented by 'True' (which equals '0') and 'False' (which equals '1')

- [SPINNAKERC\\_API spinCommandExecute](#) ([spinNodeHandle](#) hNode)  
Executes the action associated to a command node.
- [SPINNAKERC\\_API spinCommandIsDone](#) ([spinNodeHandle](#) hNode, [bool8\\_t](#) \*pbValue)  
Retrieves whether or not the action of a command node has completed.
- [SPINNAKERC\\_API spinCategoryGetNumFeatures](#) ([spinNodeHandle](#) hNode, [size\\_t](#) \*pValue)  
Retrieves the number of a features (or child nodes) of a category node.
- [SPINNAKERC\\_API spinCategoryGetFeatureByIndex](#) ([spinNodeHandle](#) hNode, [size\\_t](#) index, [spinNodeHandle](#) \*phFeature)  
Retrieves a node from a category node using an index.
- [SPINNAKERC\\_API spinRegisterGet](#) ([spinNodeHandle](#) hNode, [uint8\\_t](#) \*pBuf, [int64\\_t](#) length)  
Retrieves the value of a register node.
- [SPINNAKERC\\_API spinRegisterGetEx](#) ([spinNodeHandle](#) hNode, [bool8\\_t](#) bVerify, [bool8\\_t](#) bIgnoreCache, [uint8\\_t](#) \*pBuf, [int64\\_t](#) length)  
Retrieves the value of a register node; manually set whether to verify the node and whether to ignore the cache.
- [SPINNAKERC\\_API spinRegisterGetAddress](#) ([spinNodeHandle](#) hNode, [int64\\_t](#) \*pAddress)  
Retrieves the address of a register node.
- [SPINNAKERC\\_API spinRegisterGetLength](#) ([spinNodeHandle](#) hNode, [int64\\_t](#) \*pLength)  
Retrieves the length (in bytes) of the value of a register node.
- [SPINNAKERC\\_API spinRegisterSet](#) ([spinNodeHandle](#) hNode, [const uint8\\_t](#) \*pBuf, [int64\\_t](#) length)  
Sets the value of a register node.
- [SPINNAKERC\\_API spinRegisterSetEx](#) ([spinNodeHandle](#) hNode, [bool8\\_t](#) bVerify, [const uint8\\_t](#) \*pBuf, [int64\\_t](#) length)  
Sets the value of a register node; manually set whether to verify the node.
- [SPINNAKERC\\_API spinRegisterSetReference](#) ([spinNodeHandle](#) hNode, [spinNodeHandle](#) hRef)  
Uses a second node as a reference for a register node.

## 6.7.1 Function Documentation

### 6.7.1.1 spinBooleanGetValue()

```
SPINNAKERC_API spinBooleanGetValue (
    spinNodeHandle hNode,
    bool8_t * pbValue )
```

Retrieves the value of a boolean node; boolean values are represented by 'True' (which equals '0') and 'False' (which equals '1')

See also

[spinError](#)

Parameters

|               |                                                    |
|---------------|----------------------------------------------------|
| <i>hNode</i>  | The boolean node of the value to read              |
| <i>pValue</i> | The boolean pointer in which the value is returned |

**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.7.1.2 spinBooleanSetValue()**

```
SPINNAKERC_API spinBooleanSetValue (
    spinNodeHandle hNode,
    bool8_t value )
```

Sets the value of a boolean node; boolean values are represented by 'True' (which equals '0') and 'False' (which equals '1')

**See also**

spinError

**Parameters**

|              |                                           |
|--------------|-------------------------------------------|
| <i>hNode</i> | The boolean node having its value changed |
| <i>value</i> | The boolean value to set                  |

**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.7.1.3 spinCategoryGetFeatureByIndex()**

```
SPINNAKERC_API spinCategoryGetFeatureByIndex (
    spinNodeHandle hNode,
    size_t index,
    spinNodeHandle * phFeature )
```

Retrieves a node from a category node using an index.

**See also**

spinError

**Parameters**

|                  |                                                               |
|------------------|---------------------------------------------------------------|
| <i>hNode</i>     | The category node of the node to retrieve                     |
| <i>index</i>     | The index of the feature node                                 |
| <i>phFeature</i> | The node handle pointer in which the feature node is returned |

**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.7.1.4 spinCategoryGetNumFeatures()**

```
SPINNAKERC_API spinCategoryGetNumFeatures (
    spinNodeHandle hNode,
    size_t * pValue )
```

Retrieves the number of a features (or child nodes) or a category node.

**See also**

spinError

**Parameters**

|               |                                                                          |
|---------------|--------------------------------------------------------------------------|
| <i>hNode</i>  | The category node where the features to be counted are                   |
| <i>pValue</i> | The unsigned integer pointer in which the number of features is returned |

**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.7.1.5 spinCommandExecute()**

```
SPINNAKERC_API spinCommandExecute (
    spinNodeHandle hNode )
```

Executes the action associated to a command node.

**See also**

spinError

**Parameters**

|              |                             |
|--------------|-----------------------------|
| <i>hNode</i> | The command node to execute |
|--------------|-----------------------------|

**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

### 6.7.1.6 spinCommandIsDone()

```
SPINNAKERC_API spinCommandIsDone (
    spinNodeHandle hNode,
    bool8_t * pbValue )
```

Retrieves whether or not the action of a command node has completed.

See also

[spinError](#)

#### Parameters

|               |                                                                        |
|---------------|------------------------------------------------------------------------|
| <i>hNode</i>  | The command node to check                                              |
| <i>pValue</i> | The boolean pointer to return whether or not the command has completed |

#### Returns

[spinError](#) The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

### 6.7.1.7 spinEnumerationEntryGetEnumValue()

```
SPINNAKERC_API spinEnumerationEntryGetEnumValue (
    spinNodeHandle hNode,
    size_t * pValue )
```

Retrieves the enum value (as an integer) of an entry node; note that enumeration entry int and enum values are different - int values defined on camera, enum values found in [SpinnakerDefsC.h](#).

See also

[spinEnumerationSetEnumValue\(\)](#)

[spinError](#)

#### Parameters

|               |                                                                               |
|---------------|-------------------------------------------------------------------------------|
| <i>hNode</i>  | The entry node of the enum value to retrieve                                  |
| <i>pValue</i> | The unsigned integer pointer in which the enum value of the entry is returned |

#### Returns

[spinError](#) The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

### 6.7.1.8 spinEnumerationEntryGetIntValue()

```
SPINNAKERC_API spinEnumerationEntryGetIntValue (
    spinNodeHandle hNode,
    int64_t * pValue )
```

Retrieves the integer value of an entry node; note that enumeration entry int and enum values are different - int values defined on camera, enum values found in [SpinnakerDefsC.h](#).

See also

[spinEnumerationSetIntValue\(\)](#)  
[spinError](#)

#### Parameters

|               |                                                                         |
|---------------|-------------------------------------------------------------------------|
| <i>hNode</i>  | The entry node of the integer value to retrieve                         |
| <i>pValue</i> | The integer pointer in which the integer value of the entry is returned |

#### Returns

[spinError](#) The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

### 6.7.1.9 spinEnumerationEntryGetSymbolic()

```
SPINNAKERC_API spinEnumerationEntryGetSymbolic (
    spinNodeHandle hNode,
    char * pBuf,
    size_t * pBufLen )
```

Retrieves the symbolic of an entry node as a c-string.

See also

[spinError](#)

#### Parameters

|                |                                                                                                                     |
|----------------|---------------------------------------------------------------------------------------------------------------------|
| <i>hNode</i>   | The entry node of the symbolic to retrieve                                                                          |
| <i>pBuf</i>    | The c-string character buffer in which the symbolic of the entry node is returned                                   |
| <i>pBufLen</i> | The unsigned integer pointer in which the length of the c-string is returned; the input value is the maximum length |

#### Returns

[spinError](#) The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

#### 6.7.1.10 spinEnumerationGetCurrentEntry()

```
SPINNAKERC_API spinEnumerationGetCurrentEntry (
    spinNodeHandle hNode,
    spinNodeHandle * phEntry )
```

Retrieves the currently selected entry node from an enum node.

##### See also

spinError

##### Parameters

|                |                                                                     |
|----------------|---------------------------------------------------------------------|
| <i>hNode</i>   | The enum node from which the current entry node is retrieved        |
| <i>phEntry</i> | The node handle pointer in which the current entry node is returned |

##### Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

#### 6.7.1.11 spinEnumerationGetEntryByIndex()

```
SPINNAKERC_API spinEnumerationGetEntryByIndex (
    spinNodeHandle hNode,
    size_t index,
    spinNodeHandle * phEntry )
```

Retrieves an entry node from an enum node using an index.

##### See also

spinError

##### Parameters

|                |                                                             |
|----------------|-------------------------------------------------------------|
| <i>hNode</i>   | The enum node from which the entry node is retrieved        |
| <i>index</i>   | The index of the entry node                                 |
| <i>phEntry</i> | The node handle pointer in which the entry node is returned |

##### Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

#### 6.7.1.12 spinEnumerationGetEntryByName()

```
SPINNAKERC_API spinEnumerationGetEntryByName (
    spinNodeHandle hNode,
    const char * pName,
    spinNodeHandle * phEntry )
```

Retrieves an entry node from an enum node using the entry's symbolic.

##### See also

spinError

##### Parameters

|                |                                                             |
|----------------|-------------------------------------------------------------|
| <i>hNode</i>   | The enum node from which the entry node is retrieved        |
| <i>pName</i>   | The name of the entry node                                  |
| <i>phEntry</i> | The node handle pointer in which the entry node is returned |

##### Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

#### 6.7.1.13 spinEnumerationGetNumEntries()

```
SPINNAKERC_API spinEnumerationGetNumEntries (
    spinNodeHandle hNode,
    size_t * pValue )
```

Retrieves the number of entries of an enum node.

##### See also

spinError

##### Parameters

|               |                                                                         |
|---------------|-------------------------------------------------------------------------|
| <i>hNode</i>  | The enum node where the entries to be counted are                       |
| <i>pValue</i> | The unsigned integer pointer in which the number of entries is returned |

##### Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error



#### 6.7.1.14 spinEnumerationSetEnumValue()

```
SPINNAKERC_API spinEnumerationSetEnumValue (
    spinNodeHandle hNode,
    size_t value )
```

Sets a new entry using its enum; note that enumeration entry int and enum values are different - int values defined on camera, enum values found in [SpinnakerDefsC.h](#).

See also

[spinEnumerationEntryGetEnumValue\(\)](#)  
[spinError](#)

##### Parameters

|              |                                                                                                       |
|--------------|-------------------------------------------------------------------------------------------------------|
| <i>hNode</i> | The enum node have its entry changed                                                                  |
| <i>value</i> | The enum value of the entry node to set; this corresponds to its integer value created in the library |

##### Returns

[spinError](#) The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

#### 6.7.1.15 spinEnumerationSetIntValue()

```
SPINNAKERC_API spinEnumerationSetIntValue (
    spinNodeHandle hNode,
    int64_t value )
```

Sets a new entry using its integer value retrieved from a call to [spinEnumerationEntryGetIntValue\(\)](#); note that enumeration entry int and enum values are different - int values defined on camera, enum values found in [SpinnakerDefsC.h](#).

See also

[spinEnumerationEntryGetIntValue\(\)](#)  
[spinError](#)

##### Parameters

|              |                                                                                                          |
|--------------|----------------------------------------------------------------------------------------------------------|
| <i>hNode</i> | The enum node having its entry changed                                                                   |
| <i>value</i> | The integer value of the entry node to set; this corresponds to the integer value internal to the camera |

##### Returns

[spinError](#) The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

#### 6.7.1.16 spinFloatGetMax()

```
SPINNAKERC_API spinFloatGetMax (
    spinNodeHandle hNode,
    double * pValue )
```

Retrieves the maximum value of a float node; all potential values must be lesser than or equal to the maximum.

See also

spinError

##### Parameters

|               |                                                           |
|---------------|-----------------------------------------------------------|
| <i>hNode</i>  | The float node of the maximum value to retrieve           |
| <i>pValue</i> | The double pointer in which the maximum value is returned |

##### Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

#### 6.7.1.17 spinFloatGetMin()

```
SPINNAKERC_API spinFloatGetMin (
    spinNodeHandle hNode,
    double * pValue )
```

Retrieves the minimum value of a float node; all potential values must be greater than or equal to the minimum.

See also

spinError

##### Parameters

|               |                                                           |
|---------------|-----------------------------------------------------------|
| <i>hNode</i>  | The float node of the minimum value to retrieve           |
| <i>pValue</i> | The double pointer in which the minimum value is returned |

##### Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

#### 6.7.1.18 spinFloatGetRepresentation()

```
SPINNAKERC_API spinFloatGetRepresentation (
    spinNodeHandle hNode,
    spinRepresentation * pValue )
```

Retrieves the numerical representation of the value of a node; i.e.

linear, logarithmic, hexadecimal, MAC address, etc.

See also

`spinError`

Parameters

|               |                                                                                           |
|---------------|-------------------------------------------------------------------------------------------|
| <i>hNode</i>  | The float node of the numerical representation to retrieve                                |
| <i>pValue</i> | The representation enum pointer in which the type of numerical representation is returned |

Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

#### 6.7.1.19 `spinFloatGetUnit()`

```
SPINNAKERC_API spinFloatGetUnit (  
    spinNodeHandle hNode,  
    char * pBuf,  
    size_t * pBufLen )
```

Retrieves the units of the float node value.

See also

`spinError`

Parameters

|                |                                                                                                                     |
|----------------|---------------------------------------------------------------------------------------------------------------------|
| <i>hNode</i>   | The float node of the units to retrieve                                                                             |
| <i>pBuf</i>    | The c-string character buffer in which the value units are returned                                                 |
| <i>pBufLen</i> | The unsigned integer pointer in which the length of the c-string is returned; the input value is the maximum length |

Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

#### 6.7.1.20 `spinFloatGetValue()`

```
SPINNAKERC_API spinFloatGetValue (  
    spinNodeHandle hNode,  
    double * pValue )
```

Retrieves the value of a float node.

**See also**

spinError

**Parameters**

|               |                                                   |
|---------------|---------------------------------------------------|
| <i>hNode</i>  | The float node of the value to read               |
| <i>pValue</i> | The double pointer in which the value is returned |

**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.7.1.21 spinFloatGetValueEx()**

```
SPINNAKERC_API spinFloatGetValueEx (
    spinNodeHandle hNode,
    bool8_t bVerify,
    double * pValue )
```

Retrieves the value of a float node; manually set whether to verify the node.

**See also**

spinError

**Parameters**

|               |                                                   |
|---------------|---------------------------------------------------|
| <i>hNode</i>  | The float node of the value to read               |
| <i>pValue</i> | The double pointer in which the value is returned |

**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.7.1.22 spinFloatSetValue()**

```
SPINNAKERC_API spinFloatSetValue (
    spinNodeHandle hNode,
    double value )
```

Sets the value of a float node.

**See also**

spinError

## Parameters

|              |                                         |
|--------------|-----------------------------------------|
| <i>hNode</i> | The float node having its value changed |
| <i>value</i> | The float value to set                  |

## Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.7.1.23 spinFloatSetValueEx()**

```
SPINNAKERC_API spinFloatSetValueEx (  
    spinNodeHandle hNode,  
    bool8_t bVerify,  
    double value )
```

Sets the value of a float node; manually set whether to verify the node.

## See also

spinError

## Parameters

|                |                                           |
|----------------|-------------------------------------------|
| <i>hNode</i>   | The float node having its value changed   |
| <i>bVerify</i> | The boolean of whether to verify the node |
| <i>value</i>   | The float value to set                    |

## Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.7.1.24 spinIntegerGetInc()**

```
SPINNAKERC_API spinIntegerGetInc (  
    spinNodeHandle hNode,  
    int64_t * pValue )
```

Retrieves the increment of an integer node; all possible values must be divisible by the increment.

## See also

spinError

**Parameters**

|               |                                                        |
|---------------|--------------------------------------------------------|
| <i>hNode</i>  | The integer node of the increment to retrieve          |
| <i>pValue</i> | The integer pointer in which the increment is returned |

**Returns**

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

**6.7.1.25 spinIntegerGetMax()**

```
SPINNAKERC_API spinIntegerGetMax (  
    spinNodeHandle hNode,  
    int64_t * pValue )
```

Retrieves the maximum value of an integer node; all potential values must be lesser than or equal to the maximum.

**See also**

`spinError`

**Parameters**

|               |                                                            |
|---------------|------------------------------------------------------------|
| <i>hNode</i>  | The integer node of the maximum value to retrieve          |
| <i>pValue</i> | The integer pointer in which the maximum value is returned |

**Returns**

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

**6.7.1.26 spinIntegerGetMin()**

```
SPINNAKERC_API spinIntegerGetMin (  
    spinNodeHandle hNode,  
    int64_t * pValue )
```

Retrieves the minimum value of an integer node; all potential values must be greater than or equal to the minimum.

**See also**

`spinError`

## Parameters

|               |                                                            |
|---------------|------------------------------------------------------------|
| <i>hNode</i>  | The integer node of the minimum value to retrieve          |
| <i>pValue</i> | The integer pointer in which the minimum value is returned |

## Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.7.1.27 spinIntegerGetRepresentation()**

```
SPINNAKERC_API spinIntegerGetRepresentation (
    spinNodeHandle hNode,
    spinRepresentation * pValue )
```

Retrieves the numerical representation of the value of a node; i.e.

linear, logarithmic, hexadecimal, MAC address, etc.

## See also

spinError

## Parameters

|               |                                                                                           |
|---------------|-------------------------------------------------------------------------------------------|
| <i>hNode</i>  | The integer node of the numerical representation to retrieve                              |
| <i>pValue</i> | The representation enum pointer in which the type of numerical representation is returned |

## Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.7.1.28 spinIntegerGetValue()**

```
SPINNAKERC_API spinIntegerGetValue (
    spinNodeHandle hNode,
    int64_t * pValue )
```

Retrieves the value of an integer node.

## See also

spinError

**Parameters**

|               |                                                    |
|---------------|----------------------------------------------------|
| <i>hNode</i>  | The integer node of the value to read              |
| <i>pValue</i> | The integer pointer in which the value is returned |

**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.7.1.29 spinIntegerGetValueEx()**

```
SPINNAKERC_API spinIntegerGetValueEx (  
    spinNodeHandle hNode,  
    bool8_t bVerify,  
    int64_t * pValue )
```

Retrieves the value of an integer node; manually set whether to verify the node.

**See also**

spinError

**Parameters**

|                |                                                    |
|----------------|----------------------------------------------------|
| <i>hNode</i>   | The integer node of the value to read              |
| <i>bVerify</i> | The boolean of whether to verify the node          |
| <i>pValue</i>  | The integer pointer in which the value is returned |

**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.7.1.30 spinIntegerSetValue()**

```
SPINNAKERC_API spinIntegerSetValue (  
    spinNodeHandle hNode,  
    int64_t value )
```

Sets the value of an integer node.

**See also**

spinError



## Parameters

|              |                                           |
|--------------|-------------------------------------------|
| <i>hNode</i> | The integer node having its value changed |
| <i>value</i> | The integer value to set                  |

## Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.7.1.31 spinIntegerSetValueEx()**

```
SPINNAKERC_API spinIntegerSetValueEx (  
    spinNodeHandle hNode,  
    bool8_t bVerify,  
    int64_t value )
```

Sets the value of an integer node; manually set whether to verify the node.

## See also

spinError

## Parameters

|                |                                           |
|----------------|-------------------------------------------|
| <i>hNode</i>   | The integer node having its value changed |
| <i>bVerify</i> | The boolean of whether to verify the node |
| <i>value</i>   | The integer value to set                  |

## Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.7.1.32 spinNodeDeregisterCallback()**

```
SPINNAKERC_API spinNodeDeregisterCallback (  
    spinNodeHandle hNode,  
    spinNodeCallbackHandle hCb )
```

Unregisters a callback from a node.

## See also

spinError

## Parameters

|              |                                                |
|--------------|------------------------------------------------|
| <i>hNode</i> | The node from which to unregister the callback |
| <i>hCb</i>   | The callback handle to unregister              |

## Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.7.1.33 spinNodeFromString()**

```
SPINNAKERC_API spinNodeFromString (
    spinNodeHandle hNode,
    const char * pBuf )
```

Sets the value of any node type from a c-string; it is important to ensure that the value of the c-string is appropriate to the node type.

## See also

spinError

## Parameters

|              |                                   |
|--------------|-----------------------------------|
| <i>hNode</i> | The node having its value changed |
| <i>pBuf</i>  | The c-string of the value to set  |

## Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.7.1.34 spinNodeFromStringEx()**

```
SPINNAKERC_API spinNodeFromStringEx (
    spinNodeHandle hNode,
    bool8_t bVerify,
    const char * pBuf )
```

Sets the value of any node type from a c-string; manually set whether to verify the node; ensure the value of the c-string is appropriate to the node type.

## See also

spinError

## Parameters

|                |                                           |
|----------------|-------------------------------------------|
| <i>hNode</i>   | The node having its value changed         |
| <i>bVerify</i> | The boolean of whether to verify the node |
| <i>pBuf</i>    | The c-string of the value to set          |

## Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.7.1.35 spinNodeGetAccessMode()**

```
SPINNAKERC_API spinNodeGetAccessMode (
    spinNodeHandle hNode,
    spinAccessMode * pAccessMode )
```

Retrieves the access mode of a node (as an enum, spinAccessMode)

## See also

spinError  
spinAccessMode

## Parameters

|                    |                                                                   |
|--------------------|-------------------------------------------------------------------|
| <i>hNode</i>       | The node of the access mode to retrieve                           |
| <i>pAccessMode</i> | The access mode enum pointer in which the access mode is returned |

## Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.7.1.36 spinNodeGetCachingMode()**

```
SPINNAKERC_API spinNodeGetCachingMode (
    spinNodeHandle hNode,
    spinCachingMode * pCachingMode )
```

Retrieves the caching mode of a node (as an enum, spinCachingMode)

## See also

spinError  
spinCachingMode

## Parameters

|                     |                                                                     |
|---------------------|---------------------------------------------------------------------|
| <i>hNode</i>        | The node of the caching mode to retrieve                            |
| <i>pCachingMode</i> | The caching mode enum pointer in which the caching mode is returned |

## Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

**6.7.1.37 spinNodeGetDescription()**

```
SPINNAKERC_API spinNodeGetDescription (
    spinNodeHandle hNode,
    char * pBuf,
    size_t * pBufLen )
```

Retrieves a longer description of a node.

## See also

`spinError`

## Parameters

|                |                                                                                                                     |
|----------------|---------------------------------------------------------------------------------------------------------------------|
| <i>hNode</i>   | The node of the description to retrieve                                                                             |
| <i>pBuf</i>    | The c-string character buffer in which the longer descrtion of the node is returned                                 |
| <i>pBufLen</i> | The unsigned integer pointer in which the length of the c-string is returned; the input value is the maximum length |

## Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

**6.7.1.38 spinNodeGetDisplayName()**

```
SPINNAKERC_API spinNodeGetDisplayName (
    spinNodeHandle hNode,
    char * pBuf,
    size_t * pBufLen )
```

Retrieves the display name of a node (whitespace possible)

## See also

`spinError`

## Parameters

|                |                                                                                                                     |
|----------------|---------------------------------------------------------------------------------------------------------------------|
| <i>hNode</i>   | The node of the display name to retrieve                                                                            |
| <i>pBuf</i>    | The c-string character buffer in which the display name of the node is returned                                     |
| <i>pBufLen</i> | The unsigned integer pointer in which the length of the c-string is returned; the input value is the maximum length |

## Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.7.1.39 spinNodeGetImposedAccessMode()**

```
SPINNAKERC_API spinNodeGetImposedAccessMode (
    spinNodeHandle hNode,
    spinAccessMode imposedAccessMode )
```

Retrieves the imposed access mode of a node.

## See also

spinError

## Parameters

|                          |                                                                           |
|--------------------------|---------------------------------------------------------------------------|
| <i>hNode</i>             | The node of the imposed access mode to retrieve                           |
| <i>imposedAccessMode</i> | The access mode enum pointer in which the imposed access mode is returned |

## Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.7.1.40 spinNodeGetImposedVisibility()**

```
SPINNAKERC_API spinNodeGetImposedVisibility (
    spinNodeHandle hNode,
    spinVisibility imposedVisibility )
```

Retrieves the imposed visibility of a node.

## See also

spinError

## Parameters

|                          |                                                                         |
|--------------------------|-------------------------------------------------------------------------|
| <i>hNode</i>             | The node of the visibility to impose                                    |
| <i>imposedVisibility</i> | The visibility enum pointer in which the imposed visibility is returned |

## Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.7.1.41 spinNodeGetName()**

```
SPINNAKERC_API spinNodeGetName (
    spinNodeHandle hNode,
    char * pBuf,
    size_t * pBufLen )
```

Retrieves the name of a node (no whitespace)

## See also

spinError

## Parameters

|                |                                                                                                                     |
|----------------|---------------------------------------------------------------------------------------------------------------------|
| <i>hNode</i>   | The node of the name to retrieve                                                                                    |
| <i>pBuf</i>    | The c-string character buffer in which the name of the node is returned                                             |
| <i>pBufLen</i> | The unsigned integer pointer in which the length of the c-string is returned; the input value is the maximum length |

## Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.7.1.42 spinNodeGetNameSpace()**

```
SPINNAKERC_API spinNodeGetNameSpace (
    spinNodeHandle hNode,
    spinNameSpace * pNamespace )
```

Retrieve the namespace of a node (as an enum, spinNameSpace)

## See also

spinError

spinNameSpace

## Parameters

|                   |                                                               |
|-------------------|---------------------------------------------------------------|
| <i>hNode</i>      | The node of the namespace to retrieve                         |
| <i>pNamespace</i> | The namespace enum pointer in which the namespace is returned |

## Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.7.1.43 spinNodeGetPollingTime()**

```
SPINNAKERC_API spinNodeGetPollingTime (
    spinNodeHandle hNode,
    int64_t * pPollingTime )
```

Retrieve the polling time of a node.

## See also

spinError

## Parameters

|                     |                                                           |
|---------------------|-----------------------------------------------------------|
| <i>hNode</i>        | The node of the polling time to retrieve                  |
| <i>pPollingTime</i> | The integer pointer in which the polling time is returned |

## Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.7.1.44 spinNodeGetToolTip()**

```
SPINNAKERC_API spinNodeGetToolTip (
    spinNodeHandle hNode,
    char * pBuf,
    size_t * pBufLen )
```

Retrieves a short description of a node.

## See also

spinError

## Parameters

|                |                                                                                                                     |
|----------------|---------------------------------------------------------------------------------------------------------------------|
| <i>hNode</i>   | The node of the tooltip to retrieve                                                                                 |
| <i>pBuf</i>    | The c-string character buffer in which the short description of the node is returned                                |
| <i>pBufLen</i> | The unsigned integer pointer in which the length of the c-string is returned; the input value is the maximum length |

## Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.7.1.45 spinNodeGetType()**

```
SPINNAKERC_API spinNodeGetType (
    spinNodeHandle hNode,
    spinNodeType * pType )
```

Retrieves the type of a node (as an enum, spinNodeType)

## See also

spinError  
spinNodeType

## Parameters

|              |                                                                  |
|--------------|------------------------------------------------------------------|
| <i>hNode</i> | The node of the node type to retrieve                            |
| <i>pType</i> | The node type enum pointer in which the type of node is returned |

## Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.7.1.46 spinNodeGetVisibility()**

```
SPINNAKERC_API spinNodeGetVisibility (
    spinNodeHandle hNode,
    spinVisibility * pVisibility )
```

Retrieves the recommended visibility of a node (as an enum, spinVisibility)

## See also

spinError  
spinVisibility



## Parameters

|                    |                                                                 |
|--------------------|-----------------------------------------------------------------|
| <i>hNode</i>       | The node of the visibility to retrieve                          |
| <i>pVisibility</i> | The visibility enum pointer in which the visibility is returned |

## Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.7.1.47 spinNodeInvalidateNode()**

```
SPINNAKERC_API spinNodeInvalidateNode (  
    spinNodeHandle hNode )
```

Invalidates a node in case its values may have changed, rendering it no longer valid.

## See also

spinError

## Parameters

|              |                                        |
|--------------|----------------------------------------|
| <i>hNode</i> | The node whose values may have changed |
|--------------|----------------------------------------|

## Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.7.1.48 spinNodeIsAvailable()**

```
SPINNAKERC_API spinNodeIsAvailable (  
    spinNodeHandle hNode,  
    bool8_t * pbResult )
```

Checks whether a node is available.

## See also

spinError

## Parameters

|                 |                                                                    |
|-----------------|--------------------------------------------------------------------|
| <i>hNode</i>    | The node to check                                                  |
| <i>pbResult</i> | The boolean pointer to return whether or not the node is available |

**Returns**

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

**6.7.1.49 spinNodeIsEqual()**

```
SPINNAKERC_API spinNodeIsEqual (
    spinNodeHandle hNodeFirst,
    spinNodeHandle hNodeSecond,
    bool8_t * pbResult )
```

Checks whether two nodes are equal.

**See also**

`spinError`

**Parameters**

|                    |                                                                      |
|--------------------|----------------------------------------------------------------------|
| <i>hNodeFirst</i>  | The first node to check                                              |
| <i>hNodeSecond</i> | The second node to check                                             |
| <i>pbResult</i>    | The boolean pointer to return whether or not the two nodes are equal |

**Returns**

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

**6.7.1.50 spinNodeIsImplemented()**

```
SPINNAKERC_API spinNodeIsImplemented (
    spinNodeHandle hNode,
    bool8_t * pbResult )
```

Checks whether a node is implemented.

**See also**

`spinError`

**Parameters**

|                 |                                                                      |
|-----------------|----------------------------------------------------------------------|
| <i>hNode</i>    | The node to check                                                    |
| <i>pbResult</i> | The boolean pointer to return whether or not the node is implemented |

**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.7.1.51 spinNodeIsReadable()**

```
SPINNAKERC_API spinNodeIsReadable (
    spinNodeHandle hNode,
    bool8_t * pbResult )
```

Checks whether a node is readable.

**See also**

spinError

**Parameters**

|                 |                                                                   |
|-----------------|-------------------------------------------------------------------|
| <i>hNode</i>    | The node to check                                                 |
| <i>pbResult</i> | The boolean pointer to return whether or not the node is readable |

**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.7.1.52 spinNodeIsWritable()**

```
SPINNAKERC_API spinNodeIsWritable (
    spinNodeHandle hNode,
    bool8_t * pbResult )
```

Checks whether a node is writable.

**See also**

spinError

**Parameters**

|                 |                                                                   |
|-----------------|-------------------------------------------------------------------|
| <i>hNode</i>    | The node to check                                                 |
| <i>pbResult</i> | The boolean pointer to return whether or not the node is writable |

**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

### 6.7.1.53 spinNodeMapGetNode()

```
SPINNAKERC_API spinNodeMapGetNode (
    spinNodeMapHandle hNodeMap,
    const char * pName,
    spinNodeHandle * pNode )
```

Retrieves a node from the nodemap by name.

See also

spinError

#### Parameters

|                 |                                                       |
|-----------------|-------------------------------------------------------|
| <i>hNodeMap</i> | The node map where the node is                        |
| <i>pName</i>    | The name of the node                                  |
| <i>pNode</i>    | The node handle pointer in which the node is returned |

#### Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

### 6.7.1.54 spinNodeMapGetNodeByIndex()

```
SPINNAKERC_API spinNodeMapGetNodeByIndex (
    spinNodeMapHandle hNodeMap,
    size_t index,
    spinNodeHandle * pNode )
```

Retrieves a node from the nodemap by index.

See also

spinError

#### Parameters

|                 |                                                       |
|-----------------|-------------------------------------------------------|
| <i>hNodeMap</i> | The node map where the node is                        |
| <i>index</i>    | The index of the node                                 |
| <i>pNode</i>    | The node handle pointer in which the node is returned |

**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.7.1.55 spinNodeMapGetNumNodes()**

```
SPINNAKERC_API spinNodeMapGetNumNodes (
    spinNodeMapHandle hNodeMap,
    size_t * pValue )
```

Gets the number of nodes in the map.

**See also**

spinError

**Parameters**

|                 |                                                                       |
|-----------------|-----------------------------------------------------------------------|
| <i>hNodeMap</i> | The node map where the nodes to be counted are                        |
| <i>pValue</i>   | The unsigned integer pointer in which the number of nodes is returned |

**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.7.1.56 spinNodeMapPoll()**

```
SPINNAKERC_API spinNodeMapPoll (
    spinNodeMapHandle hNodeMap,
    int64_t timestamp )
```

Fires nodes which have a polling time.

**See also**

spinError

**Parameters**

|                  |                     |
|------------------|---------------------|
| <i>hNodeMap</i>  | The nodemap to poll |
| <i>timestamp</i> | The timestamp       |

**Returns**

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

**6.7.1.57 spinNodeRegisterCallback()**

```
SPINNAKERC_API spinNodeRegisterCallback (
    spinNodeHandle hNode,
    spinNodeCallbackFunction pCbFunction,
    spinNodeCallbackHandle * phCb )
```

Registers a callback to a node.

**See also**

`spinError`

**Parameters**

|                    |                                                                                                                                                                   |
|--------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>hNode</i>       | The node on which to register the callback                                                                                                                        |
| <i>pCbFunction</i> | The function pointer of the function that will execute when the callback is triggered; must match signature "void spinNodeCallbackFunction(spinNodeHandle hNode)" |
| <i>phCb</i>        | The callback handle pointer in which the callback is returned; used to unregister callbacks                                                                       |

**Returns**

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

**6.7.1.58 spinNodeToString()**

```
SPINNAKERC_API spinNodeToString (
    spinNodeHandle hNode,
    char * pBuf,
    size_t * pBufLen )
```

Retrieves the value of any node type as a c-string.

**See also**

`spinError`

**Parameters**

|                |                                                                                                                     |
|----------------|---------------------------------------------------------------------------------------------------------------------|
| <i>hNode</i>   | The node of the value to read                                                                                       |
| <i>pBuf</i>    | The c-string character buffer in which the value of the node is returned                                            |
| <i>pBufLen</i> | The unsigned integer pointer in which the length of the c-string is returned; the input value is the maximum length |

**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.7.1.59 spinNodeToStringEx()**

```
SPINNAKERC_API spinNodeToStringEx (
    spinNodeHandle hNode,
    bool8_t bVerify,
    char * pBuf,
    size_t * pBufLen )
```

Retrieves the value of any node type as a c-string; manually set whether to verify the node.

**See also**

spinError

**Parameters**

|                |                                                                                                                     |
|----------------|---------------------------------------------------------------------------------------------------------------------|
| <i>hNode</i>   | The node of the value to read                                                                                       |
| <i>bVerify</i> | The boolean of whether to verify the node                                                                           |
| <i>pBuf</i>    | The c-string character buffer in which the value of the node is returned                                            |
| <i>pBufLen</i> | The unsigned integer pointer in which the length of the c-string is returned; the input value is the maximum length |

**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.7.1.60 spinRegisterGet()**

```
SPINNAKERC_API spinRegisterGet (
    spinNodeHandle hNode,
    uint8_t * pBuf,
    int64_t length )
```

Retrieves the value of a register node.

**See also**

spinError

**Parameters**

|               |                                                                                                                  |
|---------------|------------------------------------------------------------------------------------------------------------------|
| <i>hNode</i>  | The register node of the value to retrieve                                                                       |
| <i>pBuf</i>   | The unsigned integer buffer in which the value is returned                                                       |
| <i>length</i> | The integer pointer in which the length of the register array is returned; the input value is the maximum length |

**Returns**

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

**6.7.1.61 spinRegisterGetAddress()**

```
SPINNAKERC_API spinRegisterGetAddress (
    spinNodeHandle hNode,
    int64_t * pAddress )
```

Retrieves the address of a register node.

**See also**

`spinError`

**Parameters**

|                 |                                                      |
|-----------------|------------------------------------------------------|
| <i>hNode</i>    | The register node of the address to retrieve         |
| <i>pAddress</i> | The integer pointer in which the address is returned |

**Returns**

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

**6.7.1.62 spinRegisterGetEx()**

```
SPINNAKERC_API spinRegisterGetEx (
    spinNodeHandle hNode,
    bool8_t bVerify,
    bool8_t bIgnoreCache,
    uint8_t * pBuf,
    int64_t length )
```

Retrieves the value of a register node; manually set whether to verify the node and whether to ignore the cache.

**See also**

`spinError`

**Parameters**

|                    |                                                                                                                  |
|--------------------|------------------------------------------------------------------------------------------------------------------|
| <i>hNode</i>       | The register node of the value to retrieve                                                                       |
| <i>bVerify</i>     | The boolean of whether to verify the node                                                                        |
| <i>IgnoreCache</i> | The boolean of whether to ignore the cache                                                                       |
| <i>pBuf</i>        | The unsigned integer buffer in which the value is returned                                                       |
| <i>length</i>      | The integer pointer in which the length of the register array is returned; the input value is the maximum length |



**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.7.1.63 spinRegisterGetLength()**

```
SPINNAKERC_API spinRegisterGetLength (
    spinNodeHandle hNode,
    int64_t * pLength )
```

Retrieves the length (in bytes) of the value of a register node.

**See also**

spinError

**Parameters**

|                |                                                      |
|----------------|------------------------------------------------------|
| <i>hNode</i>   | The register node of the length to retrieve          |
| <i>pLength</i> | The integer in which the number of bytes is returned |

**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.7.1.64 spinRegisterSet()**

```
SPINNAKERC_API spinRegisterSet (
    spinNodeHandle hNode,
    const uint8_t * pBuf,
    int64_t length )
```

Sets the value of a register node.

**See also**

spinError

**Parameters**

|               |                                                 |
|---------------|-------------------------------------------------|
| <i>hNode</i>  | The register node of the value to set           |
| <i>pBuf</i>   | The unsigned integer buffer of the value to set |
| <i>length</i> | The number of bytes of the value to set         |

**Returns**

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

**6.7.1.65 spinRegisterSetEx()**

```
SPINNAKERC_API spinRegisterSetEx (
    spinNodeHandle hNode,
    bool8_t bVerify,
    const uint8_t * pBuf,
    int64_t length )
```

Sets the value of a register node; manually set whether to verify the node.

**See also**

`spinError`

**Parameters**

|                |                                                 |
|----------------|-------------------------------------------------|
| <i>hNode</i>   | The register node of the value to set           |
| <i>bVerify</i> | The boolean of whether to verify the node       |
| <i>pBuf</i>    | The unsigned integer buffer of the value to set |
| <i>length</i>  | The number of bytes of the value to set         |

**Returns**

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

**6.7.1.66 spinRegisterSetReference()**

```
SPINNAKERC_API spinRegisterSetReference (
    spinNodeHandle hNode,
    spinNodeHandle hRef )
```

Uses a second node as a reference for a register node.

**See also**

`spinError`

**Parameters**

|              |                                             |
|--------------|---------------------------------------------|
| <i>hNode</i> | The register node that houses the reference |
| <i>hRef</i>  | The reference node                          |

**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.7.1.67 spinStringGetMaxLength()**

```
SPINNAKERC_API spinStringGetMaxLength (
    spinNodeHandle hNode,
    int64_t * pValue )
```

Retrieves the maximum length of the c-string to be returned.

**See also**

spinError

**Parameters**

|               |                                                                             |
|---------------|-----------------------------------------------------------------------------|
| <i>hNode</i>  | The string node of the length to retrieve                                   |
| <i>pValue</i> | The integer pointer in which the maximum length of the c-string is returned |

**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.7.1.68 spinStringGetValue()**

```
SPINNAKERC_API spinStringGetValue (
    spinNodeHandle hNode,
    char * pBuf,
    size_t * pBufLen )
```

Retrieves the value of a string node as a c-string.

**See also**

spinError

**Parameters**

|                |                                                                                                                     |
|----------------|---------------------------------------------------------------------------------------------------------------------|
| <i>hNode</i>   | The string node of the value to read                                                                                |
| <i>pBuf</i>    | The c-string character buffer in which the value of the node is returned                                            |
| <i>pBufLen</i> | The unsigned integer pointer in which the length of the c-string is returned; the input value is the maximum length |

**Returns**

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

**6.7.1.69 spinStringGetValueEx()**

```
SPINNAKER_API spinStringGetValueEx (
    spinNodeHandle hNode,
    bool8_t bVerify,
    char * pBuf,
    size_t * pBufLen )
```

Retrieves the value of a string node as a cstring; manually set whether to verify the node.

**See also**

`spinError`

**Parameters**

|                |                                                                                                                     |
|----------------|---------------------------------------------------------------------------------------------------------------------|
| <i>hNode</i>   | The string node of the value to read                                                                                |
| <i>bVerify</i> | The boolean of whether to verify the node                                                                           |
| <i>pBuf</i>    | The c-string character buffer in which the value of the node is returned                                            |
| <i>pBufLen</i> | The unsigned integer pointer in which the length of the c-string is returned; the input value is the maximum length |

**Returns**

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

**6.7.1.70 spinStringSetValue()**

```
SPINNAKER_API spinStringSetValue (
    spinNodeHandle hNode,
    const char * pBuf )
```

Sets the value of a string node.

**See also**

`spinError`

**Parameters**

|              |                                          |
|--------------|------------------------------------------|
| <i>hNode</i> | The string node having its value changed |
| <i>pBuf</i>  | The c-string of the value to set         |

**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.7.1.71 spinStringSetValueEx()**

```
SPINNAKER_API spinStringSetValueEx (
    spinNodeHandle hNode,
    bool8_t bVerify,
    const char * pBuf )
```

Sets the value of a string node; manually set whether to verify the node.

**See also**

spinError

**Parameters**

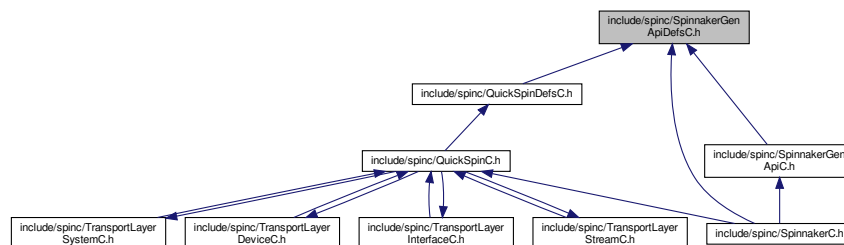
|                |                                           |
|----------------|-------------------------------------------|
| <i>hNode</i>   | The string node having its value changed  |
| <i>bVerify</i> | The boolean of whether to verify the node |
| <i>pBuf</i>    | The c-string of the value to set          |

**Returns**

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

**6.8 include/spinc/SpinnakerGenApiDefsC.h File Reference**

This graph shows which files directly or indirectly include this file:

**Typedefs**

- typedef void \* [spinNodeMapHandle](#)  
Handle for nodemap functionality.

- typedef void \* [spinNodeHandle](#)  
*Handle for node functionality.*
- typedef void \* [spinNodeCallbackHandle](#)  
*Handle for callback functionality.*
- typedef void(\* [spinNodeCallbackFunction](#)) ([spinNodeHandle](#) hNode)  
*Function signatures are used to create and trigger callbacks and events.*

## Enumerations

- enum [\\_spinNodeType](#) {  
ValueNode,  
BaseNode,  
IntegerNode,  
BooleanNode,  
FloatNode,  
CommandNode,  
StringNode,  
RegisterNode,  
EnumerationNode,  
EnumEntryNode,  
CategoryNode,  
PortNode,  
UnknownNode = -1 }
  - enum [\\_spinSign](#) {  
Signed,  
Unsigned,  
\_UndefinedSign }
  - enum [\\_spinAccessMode](#) {  
NI,  
NA,  
WO,  
RO,  
RW,  
\_UndefinedAccesMode,  
\_CycleDetectAccesMode }
  - enum [\\_spinVisibility](#) {  
Beginner = 0,  
Expert = 1,  
Guru = 2,  
Invisible = 3,  
\_UndefinedVisibility = 99 }
  - enum [\\_spinCachingMode](#) {  
NoCache,  
WriteThrough,  
WriteAround,  
\_UndefinedCachingMode }
  - enum [\\_spinRepresentation](#) {  
Linear,  
Logarithmic,  
Boolean,  
PureNumber,  
HexNumber,  
IPV4Address,  
MACAddress,  
\_UndefinedRepresentation }
- recommended representation of a node value*

- enum `_spinEndianness` {  
`BigEndian`,  
`LittleEndian`,  
`_UndefinedEndian` }  
*Endianness of a value in a register.*
- enum `_spinNameSpace` {  
`Custom`,  
`Standard`,  
`_UndefinedNameSpace` }  
*Defines if a node name is standard or custom.*
- enum `_spinStandardNameSpace` {  
`None`,  
`GEV`,  
`IIDC`,  
`CL`,  
`USB`,  
`_UndefinedStandardNameSpace` }  
*Defines from which standard namespace a node name comes from.*
- enum `_spinYesNo` {  
`Yes` = 1,  
`No` = 0,  
`_UndefinedYesNo` = 2 }  
*Defines the chices of a Yes/No alternaitve.*
- enum `_spinSlope` {  
`Increasing`,  
`Decreasing`,  
`Varying`,  
`Automatic`,  
`_UndefinedESlope` }  
*typedef for fomula type*
- enum `_spinXMLValidation` {  
`xvLoad` = 0x00000001L,  
`xvCycles` = 0x00000002L,  
`xvSFNC` = 0x00000004L,  
`xvDefault` = 0x00000000L,  
`xvAll` = 0xffffffffL,  
`_UndefinedEXMLValidation` = 0x80000000L }  
*typedef describing the different validity checks which can be performed on an XML file*
- enum `_spinDisplayNotation` {  
`fnAutomatic`,  
`fnFixed`,  
`fnScientific`,  
`_UndefinedEDisplayNotation` }  
*typedef for float notation*
- enum `_spinInterfaceType` {  
`intfIValue`,  
`intfIBase`,  
`intfIInteger`,  
`intfIBoolean`,  
`intfICommand`,  
`intfIFloat`,  
`intfIString`,  
`intfIRegister`,  
`intfICategory`,  
`intfIEnumeration`,  
`intfIEnumEntry`,  
`intfIPort` }

- typedef for interface type*
- enum `_spinLinkType` {  
`ctAllDependingNodes`,  
`ctAllTerminalNodes`,  
`ctInvalidators`,  
`ctReadingChildren`,  
`ctWritingChildren`,  
`ctDependingChildren` }
- typedef for link type*
- enum `_spinIncMode` {  
`noIncrement`,  
`fixedIncrement`,  
`listIncrement` }
- typedef for increment mode*
- enum `_spinInputDirection` {  
`idFrom`,  
`idTo`,  
`idNone` }
- typedef for link type*

## 6.8.1 Typedef Documentation

### 6.8.1.1 spinNodeCallbackFunction

```
typedef void(* spinNodeCallbackFunction) (spinNodeHandle hNode)
```

Function signatures are used to create and trigger callbacks and events.

### 6.8.1.2 spinNodeCallbackHandle

```
typedef void* spinNodeCallbackHandle
```

Handle for callback functionality.

Created by calling `spinNodeRegisterCallback()`, which requires a call to `spinNodeUnregisterCallback()` destroy.

### 6.8.1.3 spinNodeHandle

```
typedef void* spinNodeHandle
```

Handle for node functionality.

Created by calling `spinNodeMapGetNode()`. No need to release, clear, or destroy.



#### 6.8.1.4 spinNodeMapHandle

```
typedef void* spinNodeMapHandle
```

Handle for nodemap functionality.

Created by calling spinCameraGetNodemap(), [spinCameraGetTLDeviceNodeMap\(\)](#), [spinCameraGetTLStreamNodeMap\(\)](#) or [spinInterfaceGetTLNodeMap\(\)](#). No need to release, clear, or destroy.

### 6.8.2 Enumeration Type Documentation

#### 6.8.2.1 \_spinAccessMode

```
enum _spinAccessMode
```

Enumerator

|                       |  |
|-----------------------|--|
| NI                    |  |
| NA                    |  |
| WO                    |  |
| RO                    |  |
| RW                    |  |
| _UndefinedAccesMode   |  |
| _CycleDetectAccesMode |  |

#### 6.8.2.2 \_spinCachingMode

```
enum _spinCachingMode
```

Enumerator

|                       |  |
|-----------------------|--|
| NoCache               |  |
| WriteThrough          |  |
| WriteAround           |  |
| _UndefinedCachingMode |  |

#### 6.8.2.3 \_spinDisplayNotation

```
enum _spinDisplayNotation
```

typedef for float notation

## Enumerator

|                            |                                                                           |
|----------------------------|---------------------------------------------------------------------------|
| fnAutomatic                |                                                                           |
| fnFixed                    | the notation if either scientific or fixed depending on what is shorter   |
| fnScientific               | the notation is fixed, e.g. 123.4                                         |
| _UndefinedEDisplayNotation | the notation is scientific, e.g. 1.234e2<br>Object is not yet initialized |

## 6.8.2.4 \_spinEndianness

```
enum _spinEndianness
```

Endianness of a value in a register.

## Enumerator

|                  |                                |
|------------------|--------------------------------|
| BigEndian        | Register is big endian.        |
| LittleEndian     | Register is little endian.     |
| _UndefinedEndian | Object is not yet initialized. |

## 6.8.2.5 \_spinIncMode

```
enum _spinIncMode
```

typedef for increment mode

## Enumerator

|                |  |
|----------------|--|
| noIncrement    |  |
| fixedIncrement |  |
| listIncrement  |  |

## 6.8.2.6 \_spinInputDirection

```
enum _spinInputDirection
```

typedef for link type

## Enumerator

|        |                                                                                                                             |
|--------|-----------------------------------------------------------------------------------------------------------------------------|
| idFrom |                                                                                                                             |
| idTo   | Indicates a swiss knife that it is used as worker for a converter computing FROM                                            |
| idNone | Indicates a swiss knife that it is used as worker for a converter computing TO<br>SwissKnife is not used within a converter |

6.8.2.7 `_spinInterfaceType`

```
enum _spinInterfaceType
```

typedef for interface type

## Enumerator

|                               |                        |
|-------------------------------|------------------------|
| <code>intfIValue</code>       |                        |
| <code>intfIBase</code>        | IValue interface       |
| <code>intfInteger</code>      | IBase interface        |
| <code>intfIBoolean</code>     | IInteger interface     |
| <code>intfICommand</code>     | IBoolean interface     |
| <code>intfIFloat</code>       | ICommand interface     |
| <code>intfIString</code>      | IFloat interface       |
| <code>intfIRegister</code>    | IString interface      |
| <code>intfICategory</code>    | IRegister interface    |
| <code>intfIEnumeration</code> | ICategory interface    |
| <code>intfIEnumEntry</code>   | IEnumeration interface |

**Enumerator**

|           |                                         |
|-----------|-----------------------------------------|
| intfIPort | IEnumEntry interface<br>IPort interface |
|-----------|-----------------------------------------|

**6.8.2.8 \_spinLinkType**

enum `_spinLinkType`

typedef for link type

**Enumerator**

|                     |                                                                                                         |
|---------------------|---------------------------------------------------------------------------------------------------------|
| ctAllDependingNodes |                                                                                                         |
| ctAllTerminalNodes  | All nodes which will be invalidated if this node becomes invalid                                        |
| ctInvalidators      | All terminal nodes which may be written to by this node                                                 |
| ctReadingChildren   | List of references to nodes which may invalidate this node                                              |
| ctWritingChildren   | All child nodes which influence this node's AccessMode                                                  |
| ctDependingChildren | All child nodes which may be written to<br>All child nodes which will cause this node to be invalidated |

**6.8.2.9 \_spinNameSpace**

enum `_spinNameSpace`

Defines if a node name is standard or custom.

**Enumerator**

|                     |                                                |
|---------------------|------------------------------------------------|
| Custom              | name resides in custom namespace               |
| Standard            | name resides in one of the standard namespaces |
| _UndefinedNameSpace | Object is not yet initialized.                 |

### 6.8.2.10 \_spinNodeType

enum [\\_spinNodeType](#)

#### Enumerator

|                 |  |
|-----------------|--|
| ValueNode       |  |
| BaseNode        |  |
| IntegerNode     |  |
| BooleanNode     |  |
| FloatNode       |  |
| CommandNode     |  |
| StringNode      |  |
| RegisterNode    |  |
| EnumerationNode |  |
| EnumEntryNode   |  |
| CategoryNode    |  |
| PortNode        |  |
| UnknownNode     |  |

### 6.8.2.11 \_spinRepresentation

enum [\\_spinRepresentation](#)

recommended representation of a node value

#### Enumerator

|                          |                                    |
|--------------------------|------------------------------------|
| Linear                   | Slider with linear behavior.       |
| Logarithmic              | Slider with logarithmic behaviour. |
| Boolean                  | Check box.                         |
| PureNumber               | Decimal number in an edit control. |
| HexNumber                | Hex number in an edit control.     |
| IPV4Address              | IP-Address.                        |
| MACAddress               | MAC-Address.                       |
| _UndefinedRepresentation |                                    |

### 6.8.2.12 \_spinSign

enum [\\_spinSign](#)

**Enumerator**

|                |  |
|----------------|--|
| Signed         |  |
| Unsigned       |  |
| _UndefinedSign |  |

**6.8.2.13 \_spinSlope**

```
enum _spinSlope
```

```
typedef for fomula type
```

**Enumerator**

|                  |                                                                                            |
|------------------|--------------------------------------------------------------------------------------------|
| Increasing       |                                                                                            |
| Decreasing       | strictly monotonous increasing                                                             |
| Varying          | strictly monotonous decreasing                                                             |
| Automatic        | slope changes, e.g. at run-time                                                            |
| _UndefinedESlope | slope is determined automatically by probing the function<br>Object is not yet initialized |

**6.8.2.14 \_spinStandardNameSpace**

```
enum _spinStandardNameSpace
```

Defines from which standard namespace a node name comes from.

**Enumerator**

|                             |                                       |
|-----------------------------|---------------------------------------|
| None                        | name resides in custom namespace      |
| GEV                         | name resides in GigE Vision namespace |
| IIDC                        | name resides in 1394 IIDC namespace   |
| CL                          | name resides in camera link namespace |
| USB                         | name resides in USB namespace         |
| _UndefinedStandardNameSpace | Object is not yet initialized.        |

### 6.8.2.15 \_spinVisibility

enum `_spinVisibility`

Enumerator

|                      |  |
|----------------------|--|
| Beginner             |  |
| Expert               |  |
| Guru                 |  |
| Invisible            |  |
| _UndefinedVisibility |  |

### 6.8.2.16 \_spinXMLValidation

enum `_spinXMLValidation`

typedef describing the different validity checks which can be performed on an XML file

The enum values for a bitfield of length `uint32_t`

Enumerator

|                          |                                                                           |
|--------------------------|---------------------------------------------------------------------------|
| xvLoad                   |                                                                           |
| xvCycles                 | Creates a dummy node map                                                  |
| xvSFNC                   | checks for write and dependency cycles (implies xvLoad)                   |
| xvDefault                | checks for conformance with the standard feature naming convention (SFNC) |
| xvAll                    | checks performed if nothing else is said                                  |
| _UndefinedEXMLValidation | all possible checks<br>Object is not yet initialized                      |

### 6.8.2.17 \_spinYesNo

enum `_spinYesNo`

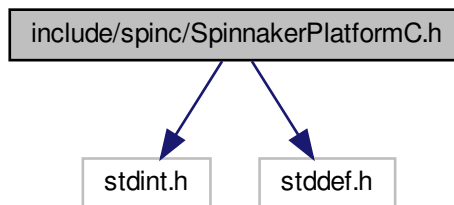
Defines the choices of a Yes/No alternative.

#### Enumerator

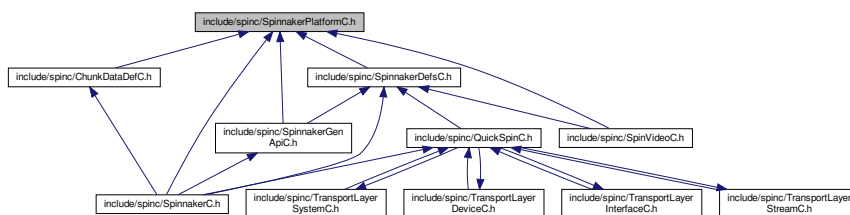
|                 |                                |
|-----------------|--------------------------------|
| Yes             | yes                            |
| No              | no                             |
| _UndefinedYesNo | Object is not yet initialized. |

## 6.9 include/spinc/SpinnakerPlatformC.h File Reference

Include dependency graph for SpinnakerPlatformC.h:



This graph shows which files directly or indirectly include this file:



## Macros

- #define `SPINNAKERC_API` `SPINC_IMPORT_EXPORT` `spinError` `SPINC_CALLTYPE`

### 6.9.1 Macro Definition Documentation

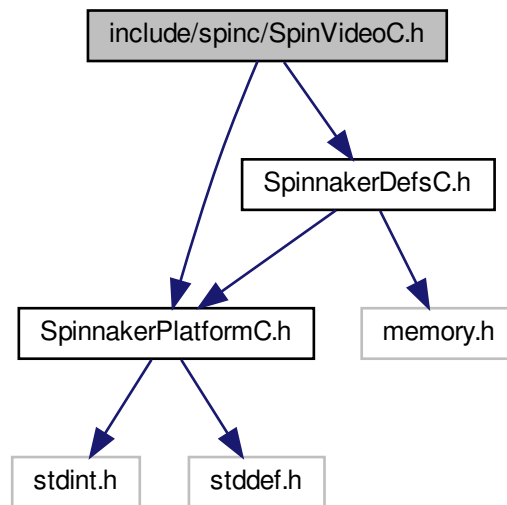


## 6.9.1.1 SPINNAKERC\_API

```
#define SPINNAKERC_API SPINC_IMPORT_EXPORT spinError SPINC_CALLTYPE
```

## 6.10 include/spinc/SpinVideoC.h File Reference

Include dependency graph for SpinVideoC.h:



## Functions

- [SPINNAKERC\\_API spinVideoOpenUncompressed](#) ([spinVideo](#) \*phSpinVideo, const char \*pName, spinAVOption option)
- [SPINNAKERC\\_API spinVideoOpenMJPEG](#) ([spinVideo](#) \*phSpinVideo, const char \*pName, spinMJPEGOption option)
- [SPINNAKERC\\_API spinVideoOpenH264](#) ([spinVideo](#) \*phSpinVideo, const char \*pName, spinH264Option option)
- [SPINNAKERC\\_API spinVideoAppend](#) ([spinVideo](#) hSpinVideo, [spinImage](#) hImage)
- [SPINNAKERC\\_API spinVideoSetMaximumFileSize](#) ([spinVideo](#) hSpinVideo, unsigned int size)  
Set the maximum file size (in megabytes) of a AVI/MP4 file.
- [SPINNAKERC\\_API spinVideoClose](#) ([spinVideo](#) hSpinVideo)

## 6.10.1 Function Documentation

#### 6.10.1.1 spinVideoAppend()

```
SPINNAKERC_API spinVideoAppend (
    spinVideo hSpinVideo,
    spinImage hImage )
```

#### 6.10.1.2 spinVideoClose()

```
SPINNAKERC_API spinVideoClose (
    spinVideo hSpinVideo )
```

#### 6.10.1.3 spinVideoOpenH264()

```
SPINNAKERC_API spinVideoOpenH264 (
    spinVideo * phSpinVideo,
    const char * pName,
    spinH264Option option )
```

#### 6.10.1.4 spinVideoOpenMJPEG()

```
SPINNAKERC_API spinVideoOpenMJPEG (
    spinVideo * phSpinVideo,
    const char * pName,
    spinMJPEGOption option )
```

#### 6.10.1.5 spinVideoOpenUncompressed()

```
SPINNAKERC_API spinVideoOpenUncompressed (
    spinVideo * phSpinVideo,
    const char * pName,
    spinAVIOption option )
```

#### 6.10.1.6 spinVideoSetMaximumFileSize()

```
SPINNAKERC_API spinVideoSetMaximumFileSize (
    spinVideo hSpinVideo,
    unsigned int size )
```

Set the maximum file size (in megabytes) of a AVI/MP4 file.

A new AVI/MP4 file is created automatically when file size limit is reached. Setting a maximum size of 0 indicates no limit on file size.

## Parameters

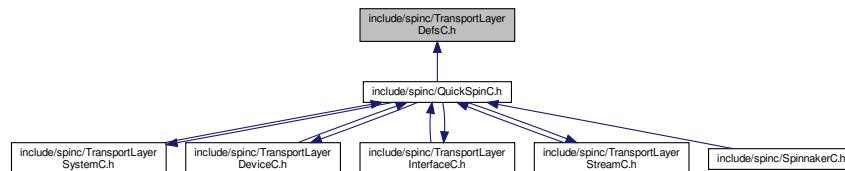
|                   |                                                |
|-------------------|------------------------------------------------|
| <i>hSpinVideo</i> | The spin video recorder to append the image to |
| <i>size</i>       | The maximum video file size in MB.             |

## Returns

spinError The error code; returns SPINNAKER\_ERR\_SUCCESS (or 0) for no error

## 6.11 include/spinc/TransportLayerDefsC.h File Reference

This graph shows which files directly or indirectly include this file:



## Enumerations

- enum `_spinTLStreamTypeEnums` {  
`StreamType_GigEVision`,  
`StreamType_CameraLink`,  
`StreamType_CameraLinkHS`,  
`StreamType_CoaXPress`,  
`StreamType_USB3Vision`,  
`StreamType_Custom`,  
`NUMSTREAMTYPE` }
- The enumeration definitions for transport layer nodes.*
- enum `_spinTLStreamBufferCountModeEnums` {  
`StreamBufferCountMode_Manual`,  
`StreamBufferCountMode_Auto`,  
`NUMSTREAMBUFFERCOUNTMODE` }
- enum `_spinTLStreamBufferHandlingModeEnums` {  
`StreamBufferHandlingMode_OldestFirst`,  
`StreamBufferHandlingMode_OldestFirstOverwrite`,  
`StreamBufferHandlingMode_NewestOnly`,  
`StreamBufferHandlingMode_NewestFirst`,  
`NUMSTREAMBUFFERHANDLINGMODE` }
- enum `_spinTLDeviceTypeEnums` {  
`DeviceType_GigEVision`,  
`DeviceType_CameraLink`,  
`DeviceType_CameraLinkHS`,  
`DeviceType_CoaXPress`,  
`DeviceType_USB3Vision`,  
`DeviceType_Custom`,  
`NUMDEVICETYPE` }

- enum `_spinTLDeviceAccessStatusEnums` {  
`DeviceAccessStatus_Unknown,`  
`DeviceAccessStatus_ReadWrite,`  
`DeviceAccessStatus_ReadOnly,`  
`DeviceAccessStatus_NoAccess,`  
`DeviceAccessStatus_Busy,`  
`DeviceAccessStatus_OpenReadWrite,`  
`DeviceAccessStatus_OpenReadOnly,`  
`NUMDEVICEACCESSSTATUS` }
- enum `_spinTLGevCCPEnums` {  
`GevCCP_EnumEntry_GevCCP_OpenAccess,`  
`GevCCP_EnumEntry_GevCCP_ExclusiveAccess,`  
`GevCCP_EnumEntry_GevCCP_ControlAccess,`  
`NUMGEVCCP` }
- enum `_spinTLGUIXMLLocationEnums` {  
`GUIXMLLocation_Device,`  
`GUIXMLLocation_Host,`  
`NUMGUIXMLLOCATION` }
- enum `_spinTLGenICamXMLLocationEnums` {  
`GenICamXMLLocation_Device,`  
`GenICamXMLLocation_Host,`  
`NUMGENICAMXMLLOCATION` }
- enum `_spinTLDeviceEndianessMechanismEnums` {  
`DeviceEndianessMechanism_Legacy,`  
`DeviceEndianessMechanism_Standard,`  
`NUMDEVICEENDIANESSMECHANISM` }
- enum `_spinTLDeviceCurrentSpeedEnums` {  
`DeviceCurrentSpeed_UnknownSpeed,`  
`DeviceCurrentSpeed_LowSpeed,`  
`DeviceCurrentSpeed_FullSpeed,`  
`DeviceCurrentSpeed_HighSpeed,`  
`DeviceCurrentSpeed_SuperSpeed,`  
`NUMDEVICECURRENTSPEED` }
- enum `_spinTLInterfaceTypeEnums` {  
`InterfaceType_GigEVision,`  
`InterfaceType_CameraLink,`  
`InterfaceType_CameraLinkHS,`  
`InterfaceType_CoaxPress,`  
`InterfaceType_USB3Vision,`  
`InterfaceType_Custom,`  
`NUMINTERFACETYPE` }
- enum `_spinTLPOEStatusEnums` {  
`POEStatus_NotSupported,`  
`POEStatus_PowerOff,`  
`POEStatus_PowerOn,`  
`NUMPOESTATUS` }
- enum `_spinTLFilterDriverStatusEnums` {  
`FilterDriverStatus_NotSupported,`  
`FilterDriverStatus_Disabled,`  
`FilterDriverStatus_Enabled,`  
`NUMFILTERDRIVERSTATUS` }
- enum `_spinTLTLTypeEnums` {  
`TLType_GigEVision,`  
`TLType_CameraLink,`  
`TLType_CameraLinkHS,`  
`TLType_CoaxPress,`  
`TLType_USB3Vision,`  
`TLType_Mixed,`

```
TLType_Custom,
NUMTLTYPE }
```

## 6.11.1 Enumeration Type Documentation

### 6.11.1.1 \_spinTLDeviceAccessStatusEnums

```
enum _spinTLDeviceAccessStatusEnums
```

< Gets the access status the transport layer Producer has on the device.

Enumerator

|                                  |                                                |
|----------------------------------|------------------------------------------------|
| DeviceAccessStatus_Unknown       | Not known to producer.                         |
| DeviceAccessStatus_ReadWrite     | Full access                                    |
| DeviceAccessStatus_ReadOnly      | Read-only access                               |
| DeviceAccessStatus_NoAccess      | Not available to connect                       |
| DeviceAccessStatus_Busy          | The device is already opened by another entity |
| DeviceAccessStatus_OpenReadWrite | Open in Read/Write mode by this GenTL host     |
| DeviceAccessStatus_OpenReadOnly  | Open in Read access mode by this GenTL host    |
| NUMDEVICEACCESSSTATUS            |                                                |

### 6.11.1.2 \_spinTLDeviceCurrentSpeedEnums

```
enum _spinTLDeviceCurrentSpeedEnums
```

< The USB Speed that the device is currently operating at.

Enumerator

|                                 |                |
|---------------------------------|----------------|
| DeviceCurrentSpeed_UnknownSpeed | Unknown-Speed. |
| DeviceCurrentSpeed_LowSpeed     | Low-Speed.     |
| DeviceCurrentSpeed_FullSpeed    | Full-Speed.    |
| DeviceCurrentSpeed_HighSpeed    | High-Speed.    |
| DeviceCurrentSpeed_SuperSpeed   | Super-Speed.   |
| NUMDEVICECURRENTSPEED           |                |

### 6.11.1.3 \_spinTLDeviceEndianessMechanismEnums

```
enum _spinTLDeviceEndianessMechanismEnums
```

< Identifies the endianness handling mode.

## Enumerator

|                                    |                                                                          |
|------------------------------------|--------------------------------------------------------------------------|
| DeviceEndiannessMechanism_Legacy   | Handling the device endianness according to GenICam Schema 1.0           |
| DeviceEndiannessMechanism_Standard | Handling the device endianness according to GenICam Schema 1.1 and later |
| NUMDEVICEENDIANESSMECHANISM        |                                                                          |

## 6.11.1.4 \_spinTLDeviceTypeEnums

```
enum _spinTLDeviceTypeEnums
```

< Transport layer type of the device.

## Enumerator

|                         |                        |
|-------------------------|------------------------|
| DeviceType_GigEVision   | GigE Vision            |
| DeviceType_CameraLink   | Camera Link            |
| DeviceType_CameraLinkHS | Camera Link High Speed |
| DeviceType_CoaXPress    | CoaXPress              |
| DeviceType_USB3Vision   | USB3 Vision            |
| DeviceType_Custom       | Custom transport layer |
| NUMDEVICETYPE           |                        |

## 6.11.1.5 \_spinTLFilterDriverStatusEnums

```
enum _spinTLFilterDriverStatusEnums
```

< Reports whether FLIR Light Weight Filter Driver is enabled or not.

## Enumerator

|                                 |                                             |
|---------------------------------|---------------------------------------------|
| FilterDriverStatus_NotSupported | Not Supported                               |
| FilterDriverStatus_Disabled     | FLIR Light Weight Filter Driver is disabled |
| FilterDriverStatus_Enabled      | FLIR Light Weight Filter Driver is enabled  |
| NUMFILTERDRIVERSTATUS           |                                             |

## 6.11.1.6 \_spinTLGenICamXMLLocationEnums

```
enum _spinTLGenICamXMLLocationEnums
```

< Sets the location to load GenICam XML.

**Enumerator**

|                           |                              |
|---------------------------|------------------------------|
| GenICamXMLLocation_Device | Load GenICam XML from device |
| GenICamXMLLocation_Host   | Load GenICam XML from host   |
| NUMGENICAMXMLLOCATION     |                              |

**6.11.1.7 \_spinTLGevCCPEnums**

```
enum _spinTLGevCCPEnums
```

< Controls the device access privilege of an application.

**Enumerator**

|                                         |                             |
|-----------------------------------------|-----------------------------|
| GevCCP_EnumEntry_GevCCP_OpenAccess      | Open access privilege.      |
| GevCCP_EnumEntry_GevCCP_ExclusiveAccess | Exclusive access privilege. |
| GevCCP_EnumEntry_GevCCP_ControlAccess   | Control access privilege.   |
| NUMGEVCCP                               |                             |

**6.11.1.8 \_spinTLGUIXMLLocationEnums**

```
enum _spinTLGUIXMLLocationEnums
```

< Sets the location to load GUI XML.

**Enumerator**

|                       |                      |
|-----------------------|----------------------|
| GUIXMLLocation_Device | Load XML from device |
| GUIXMLLocation_Host   | Load XML from host   |
| NUMGUIXMLLOCATION     |                      |

**6.11.1.9 \_spinTLInterfaceTypeEnums**

```
enum _spinTLInterfaceTypeEnums
```

< Transport layer type of the interface.

**Enumerator**

|                          |             |
|--------------------------|-------------|
| InterfaceType_GigEVision | GigE Vision |
| InterfaceType_CameraLink | Camera Link |



## Enumerator

|                            |                        |
|----------------------------|------------------------|
| InterfaceType_CameraLinkHS | Camera Link High Speed |
| InterfaceType_CoaXPress    | CoaXPress              |
| InterfaceType_USB3Vision   | USB3 Vision            |
| InterfaceType_Custom       | Custom transport layer |
| NUMINTERFACETYPE           |                        |

## 6.11.1.10 \_spinTLPOEStatusEnums

```
enum _spinTLPOEStatusEnums
```

< Reports and controls the interface's power over Ethernet status.

## Enumerator

|                        |               |
|------------------------|---------------|
| POEStatus_NotSupported | Not Supported |
| POEStatus_PowerOff     | Power is Off  |
| POEStatus_PowerOn      | Power is On   |
| NUMPOESTATUS           |               |

## 6.11.1.11 \_spinTLStreamBufferCountModeEnums

```
enum _spinTLStreamBufferCountModeEnums
```

< Controls access to setting the number of buffers used for the stream. Locked to Manual mode on 32-bit Windows due to memory constraints.

## Enumerator

|                              |                                                                                                                   |
|------------------------------|-------------------------------------------------------------------------------------------------------------------|
| StreamBufferCountMode_Manual | The number of buffers used for the stream are set by the user.                                                    |
| StreamBufferCountMode_Auto   | DEPRECATED. The number of buffers used for the stream is automatically calculated based on the device frame rate. |
| NUMSTREAMBUFFERCOUNTMODE     |                                                                                                                   |

## 6.11.1.12 \_spinTLStreamBufferHandlingModeEnums

```
enum _spinTLStreamBufferHandlingModeEnums
```

< Available buffer handling modes of this data stream:

## Enumerator

|                                               |                                                                                                                                                                                                                                                                                                                                                                |
|-----------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| StreamBufferHandlingMode_OldestFirst          | The application always gets the buffer from the head of the output buffer queue (thus, the oldest available one). If the output buffer queue is empty, the application waits for a newly acquired buffer until the timeout expires.                                                                                                                            |
| StreamBufferHandlingMode_OldestFirstOverwrite | The application always gets the buffer from the head of the output buffer queue (thus, the oldest available one). If the output buffer queue is empty, the application waits for a newly acquired buffer until the timeout expires. If a new buffer arrives it will overwrite the existing buffer from the head of the queue (behaves like a circular buffer). |
| StreamBufferHandlingMode_NewestOnly           | The application always gets the latest completed buffer (the newest one). If the Output Buffer Queue is empty, the application waits for a newly acquired buffer until the timeout expires. This buffer handling mode is typically used in a live display GUI where it is important that there is no lag between camera and display.                           |
| StreamBufferHandlingMode_NewestFirst          | The application always gets the buffer from the tail of the output buffer queue (thus, the newest available one). If the output buffer queue is empty, the application waits for a newly acquired buffer until the timeout expires.                                                                                                                            |
| NUMSTREAMBUFFERHANDLINGMODE                   |                                                                                                                                                                                                                                                                                                                                                                |

## 6.11.1.13 \_spinTLStreamTypeEnums

```
enum _spinTLStreamTypeEnums
```

The enumeration definitions for transport layer nodes.

< Stream type of the device.

## Enumerator

|                         |                        |
|-------------------------|------------------------|
| StreamType_GigEVision   | GigE Vision            |
| StreamType_CameraLink   | Camera Link            |
| StreamType_CameraLinkHS | Camera Link High Speed |
| StreamType_CoaXPress    | CoaXPress              |
| StreamType_USB3Vision   | USB3 Vision            |
| StreamType_Custom       | Custom transport layer |
| NUMSTREAMTYPE           |                        |

### 6.11.1.14 \_spinTLTLTypeEnums

enum [\\_spinTLTLTypeEnums](#)

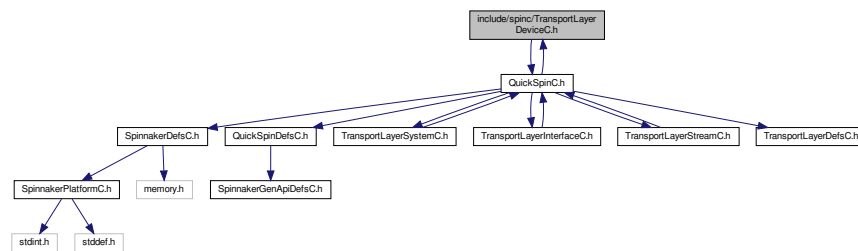
< Transport layer type of the GenTL Producer implementation.

#### Enumerator

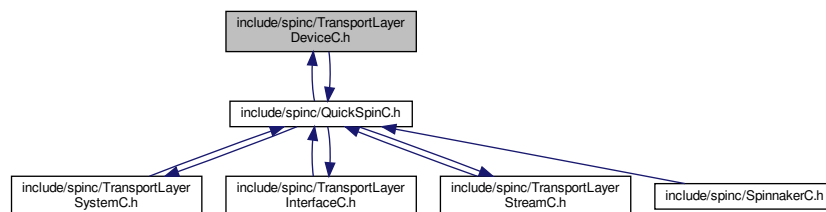
|                     |                                                                          |
|---------------------|--------------------------------------------------------------------------|
| TLType_GigEVision   | GigE Vision                                                              |
| TLType_CameraLink   | Camera Link                                                              |
| TLType_CameraLinkHS | Camera Link High Speed                                                   |
| TLType_CoaXPress    | CoaXPress                                                                |
| TLType_USB3Vision   | USB3 Vision                                                              |
| TLType_Mixed        | Different Interface modules of the GenTL Producer are of different types |
| TLType_Custom       | Custom transport layer                                                   |
| NUMTLTYPE           |                                                                          |

## 6.12 include/spinc/TransportLayerDeviceC.h File Reference

Include dependency graph for TransportLayerDeviceC.h:



This graph shows which files directly or indirectly include this file:

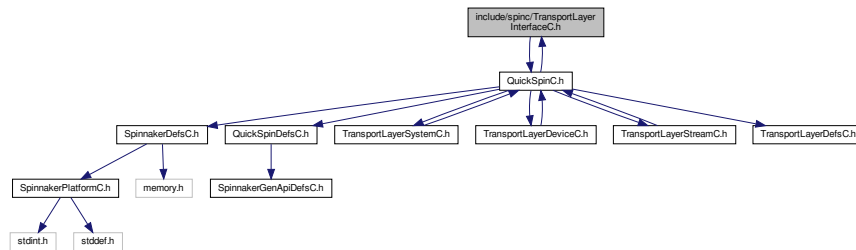


## Data Structures

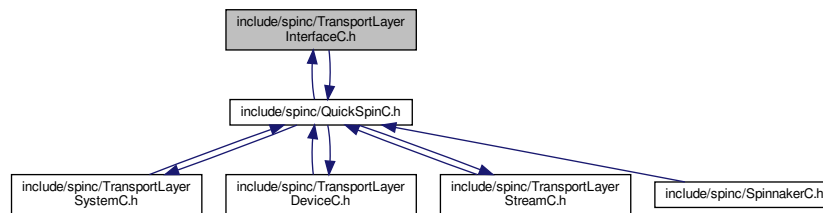
- struct [\\_quickSpinTLDevice](#)

## 6.13 include/spinc/TransportLayerInterfaceC.h File Reference

Include dependency graph for TransportLayerInterfaceC.h:



This graph shows which files directly or indirectly include this file:

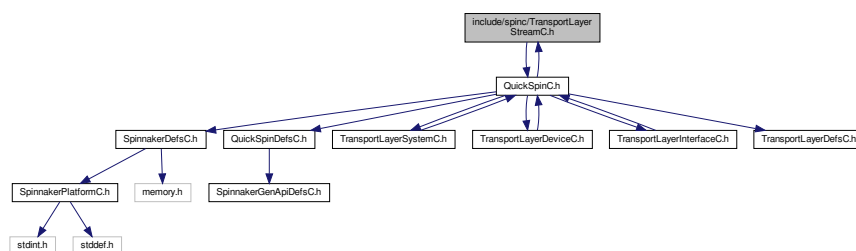


## Data Structures

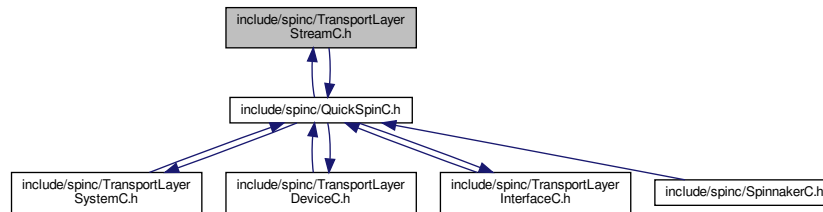
- struct [\\_quickSpinTLInterface](#)

## 6.14 include/spinc/TransportLayerStreamC.h File Reference

Include dependency graph for TransportLayerStreamC.h:



This graph shows which files directly or indirectly include this file:

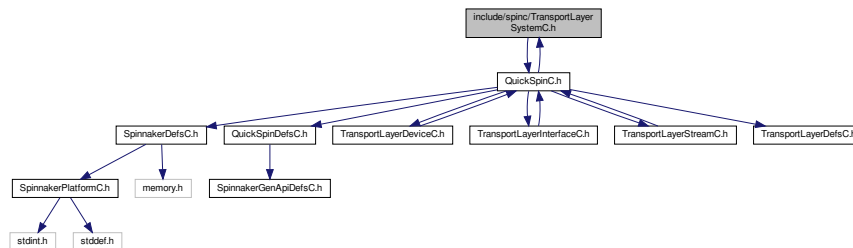


## Data Structures

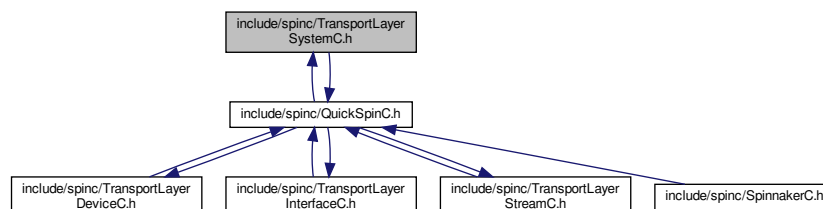
- [struct \\_quickSpinTLStream](#)

## 6.15 include/spinc/TransportLayerSystemC.h File Reference

Include dependency graph for TransportLayerSystemC.h:



This graph shows which files directly or indirectly include this file:



## Data Structures

- [struct \\_quickSpinTLSystem](#)



# Index

- [\\_CycleDetectAccessMode](#)
  - [SpinnakerGenApiDefsC.h, 457](#)
- [\\_UndefinedAccessMode](#)
  - [SpinnakerGenApiDefsC.h, 457](#)
- [\\_UndefinedCachingMode](#)
  - [SpinnakerGenApiDefsC.h, 457](#)
- [\\_UndefinedEDisplayNotation](#)
  - [SpinnakerGenApiDefsC.h, 458](#)
- [\\_UndefinedESlope](#)
  - [SpinnakerGenApiDefsC.h, 462](#)
- [\\_UndefinedEXMLValidation](#)
  - [SpinnakerGenApiDefsC.h, 463](#)
- [\\_UndefinedEndian](#)
  - [SpinnakerGenApiDefsC.h, 458](#)
- [\\_UndefinedNameSpace](#)
  - [SpinnakerGenApiDefsC.h, 460](#)
- [\\_UndefinedRepresentation](#)
  - [SpinnakerGenApiDefsC.h, 461](#)
- [\\_UndefinedSign](#)
  - [SpinnakerGenApiDefsC.h, 462](#)
- [\\_UndefinedStandardNameSpace](#)
  - [SpinnakerGenApiDefsC.h, 462](#)
- [\\_UndefinedVisibility](#)
  - [SpinnakerGenApiDefsC.h, 463](#)
- [\\_UndefinedYesNo](#)
  - [SpinnakerGenApiDefsC.h, 464](#)
- [\\_actionCommandResult, 51](#)
  - [DeviceAddress, 51](#)
  - [Status, 51](#)
- [\\_actionCommandStatus](#)
  - [SpinnakerDefsC.h, 408](#)
- [\\_quickSpin, 52](#)
  - [AasRoiEnable, 64](#)
  - [AasRoiHeight, 64](#)
  - [AasRoiOffsetX, 64](#)
  - [AasRoiOffsetY, 64](#)
  - [AasRoiWidth, 64](#)
  - [AcquisitionAbort, 65](#)
  - [AcquisitionArm, 65](#)
  - [AcquisitionBurstFrameCount, 65](#)
  - [AcquisitionFrameCount, 65](#)
  - [AcquisitionFrameRate, 65](#)
  - [AcquisitionFrameRateEnable, 65](#)
  - [AcquisitionLineRate, 65](#)
  - [AcquisitionMode, 65](#)
  - [AcquisitionResultingFrameRate, 66](#)
  - [AcquisitionStart, 66](#)
  - [AcquisitionStatus, 66](#)
  - [AcquisitionStatusSelector, 66](#)
  - [AcquisitionStop, 66](#)
  - [ActionDeviceKey, 66](#)
  - [ActionGroupKey, 66](#)
  - [ActionGroupMask, 66](#)
  - [ActionQueueSize, 67](#)
  - [ActionSelector, 67](#)
  - [ActionUnconditionalMode, 67](#)
  - [AdaptiveCompressionEnable, 67](#)
  - [AdcBitDepth, 67](#)
  - [aPAUSEMACCtrlFramesReceived, 67](#)
  - [aPAUSEMACCtrlFramesTransmitted, 67](#)
  - [AutoAlgorithmSelector, 67](#)
  - [AutoExposureControlLoopDamping, 68](#)
  - [AutoExposureControlPriority, 68](#)
  - [AutoExposureEVCompensation, 68](#)
  - [AutoExposureExposureTimeLowerLimit, 68](#)
  - [AutoExposureExposureTimeUpperLimit, 68](#)
  - [AutoExposureGainLowerLimit, 68](#)
  - [AutoExposureGainUpperLimit, 68](#)
  - [AutoExposureGreyValueLowerLimit, 68](#)
  - [AutoExposureGreyValueUpperLimit, 69](#)
  - [AutoExposureLightingMode, 69](#)
  - [AutoExposureMeteringMode, 69](#)
  - [AutoExposureTargetGreyValue, 69](#)
  - [AutoExposureTargetGreyValueAuto, 69](#)
  - [BalanceRatio, 69](#)
  - [BalanceRatioSelector, 69](#)
  - [BalanceWhiteAuto, 69](#)
  - [BalanceWhiteAutoDamping, 70](#)
  - [BalanceWhiteAutoLowerLimit, 70](#)
  - [BalanceWhiteAutoProfile, 70](#)
  - [BalanceWhiteAutoUpperLimit, 70](#)
  - [BinningHorizontal, 70](#)
  - [BinningHorizontalMode, 70](#)
  - [BinningSelector, 70](#)
  - [BinningVertical, 70](#)
  - [BinningVerticalMode, 71](#)
  - [BlackLevel, 71](#)
  - [BlackLevelAuto, 71](#)
  - [BlackLevelAutoBalance, 71](#)
  - [BlackLevelClampingEnable, 71](#)
  - [BlackLevelRaw, 71](#)
  - [BlackLevelSelector, 71](#)
  - [ChunkBlackLevel, 71](#)
  - [ChunkBlackLevelSelector, 72](#)
  - [ChunkCounterSelector, 72](#)
  - [ChunkCounterValue, 72](#)
  - [ChunkCRC, 72](#)
  - [ChunkEnable, 72](#)

- ChunkEncoderSelector, 72
- ChunkEncoderStatus, 72
- ChunkEncoderValue, 72
- ChunkExposureEndLineStatusAll, 73
- ChunkExposureTime, 73
- ChunkExposureTimeSelector, 73
- ChunkFrameID, 73
- ChunkGain, 73
- ChunkGainSelector, 73
- ChunkHeight, 73
- ChunkImage, 73
- ChunkImageComponent, 74
- ChunkInferenceBoundingBoxResult, 74
- ChunkInferenceConfidence, 74
- ChunkInferenceFrameID, 74
- ChunkInferenceResult, 74
- ChunkLinePitch, 74
- ChunkLineStatusAll, 74
- ChunkModeActive, 74
- ChunkOffsetX, 75
- ChunkOffsetY, 75
- ChunkPartSelector, 75
- ChunkPixelDynamicRangeMax, 75
- ChunkPixelDynamicRangeMin, 75
- ChunkPixelFormat, 75
- ChunkRegionID, 75
- ChunkScan3dAxisMax, 75
- ChunkScan3dAxisMin, 76
- ChunkScan3dCoordinateOffset, 76
- ChunkScan3dCoordinateReferenceSelector, 76
- ChunkScan3dCoordinateReferenceValue, 76
- ChunkScan3dCoordinateScale, 76
- ChunkScan3dCoordinateSelector, 76
- ChunkScan3dCoordinateSystem, 76
- ChunkScan3dCoordinateSystemReference, 76
- ChunkScan3dCoordinateTransformSelector, 77
- ChunkScan3dDistanceUnit, 77
- ChunkScan3dInvalidDataFlag, 77
- ChunkScan3dInvalidDataValue, 77
- ChunkScan3dOutputMode, 77
- ChunkScan3dTransformValue, 77
- ChunkScanLineSelector, 77
- ChunkSelector, 77
- ChunkSequencerSetActive, 78
- ChunkSerialData, 78
- ChunkSerialDataLength, 78
- ChunkSerialReceiveOverflow, 78
- ChunkSourceID, 78
- ChunkStreamChannelID, 78
- ChunkTimerSelector, 78
- ChunkTimerValue, 78
- ChunkTimestamp, 79
- ChunkTimestampLatchValue, 79
- ChunkTransferBlockID, 79
- ChunkTransferQueueCurrentBlockCount, 79
- ChunkTransferStreamID, 79
- ChunkWidth, 79
- CIConfiguration, 79
- CITimeSlotsCount, 79
- ColorTransformationEnable, 80
- ColorTransformationSelector, 80
- ColorTransformationValue, 80
- ColorTransformationValueSelector, 80
- CompressionRatio, 80
- CounterDelay, 80
- CounterDuration, 80
- CounterEventActivation, 80
- CounterEventSource, 81
- CounterReset, 81
- CounterResetActivation, 81
- CounterResetSource, 81
- CounterSelector, 81
- CounterStatus, 81
- CounterTriggerActivation, 81
- CounterTriggerSource, 81
- CounterValue, 82
- CounterValueAtReset, 82
- CxpConnectionSelector, 82
- CxpConnectionTestErrorCount, 82
- CxpConnectionTestMode, 82
- CxpConnectionTestPacketCount, 82
- CxpLinkConfiguration, 82
- CxpLinkConfigurationPreferred, 82
- CxpLinkConfigurationStatus, 83
- CxpPoCxpAuto, 83
- CxpPoCxpStatus, 83
- CxpPoCxpTripReset, 83
- CxpPoCxpTurnOff, 83
- DecimationHorizontal, 83
- DecimationHorizontalMode, 83
- DecimationSelector, 83
- DecimationVertical, 84
- DecimationVerticalMode, 84
- DefectCorrectionMode, 84
- DefectCorrectStaticEnable, 84
- DefectTableApply, 84
- DefectTableCoordinateX, 84
- DefectTableCoordinateY, 84
- DefectTableFactoryRestore, 84
- DefectTableIndex, 85
- DefectTablePixelCount, 85
- DefectTableSave, 85
- Deinterlacing, 85
- DeviceCharacterSet, 85
- DeviceClockFrequency, 85
- DeviceClockSelector, 85
- DeviceConnectionSelector, 85
- DeviceConnectionSpeed, 86
- DeviceConnectionStatus, 86
- DeviceEventChannelCount, 86
- DeviceFamilyName, 86
- DeviceFeaturePersistenceEnd, 86
- DeviceFeaturePersistenceStart, 86
- DeviceFirmwareVersion, 86
- DeviceGenCPVersionMajor, 86
- DeviceGenCPVersionMinor, 87



DeviceID, [87](#)  
DeviceIndicatorMode, [87](#)  
DeviceLinkBandwidthReserve, [87](#)  
DeviceLinkCommandTimeout, [87](#)  
DeviceLinkConnectionCount, [87](#)  
DeviceLinkCurrentThroughput, [87](#)  
DeviceLinkHeartbeatMode, [87](#)  
DeviceLinkHeartbeatTimeout, [88](#)  
DeviceLinkSelector, [88](#)  
DeviceLinkSpeed, [88](#)  
DeviceLinkThroughputLimit, [88](#)  
DeviceLinkThroughputLimitMode, [88](#)  
DeviceManifestEntrySelector, [88](#)  
DeviceManifestPrimaryURL, [88](#)  
DeviceManifestSchemaMajorVersion, [88](#)  
DeviceManifestSchemaMinorVersion, [89](#)  
DeviceManifestSecondaryURL, [89](#)  
DeviceManifestXMLMajorVersion, [89](#)  
DeviceManifestXMLMinorVersion, [89](#)  
DeviceManifestXMLSubMinorVersion, [89](#)  
DeviceManufacturerInfo, [89](#)  
DeviceMaxThroughput, [89](#)  
DeviceModelName, [89](#)  
DevicePowerSupplySelector, [90](#)  
DeviceRegistersCheck, [90](#)  
DeviceRegistersEndianness, [90](#)  
DeviceRegistersStreamingEnd, [90](#)  
DeviceRegistersStreamingStart, [90](#)  
DeviceRegistersValid, [90](#)  
DeviceReset, [90](#)  
DeviceScanType, [90](#)  
DeviceSerialNumber, [91](#)  
DeviceSerialPortBaudRate, [91](#)  
DeviceSerialPortSelector, [91](#)  
DeviceSFNCVersionMajor, [91](#)  
DeviceSFNCVersionMinor, [91](#)  
DeviceSFNCVersionSubMinor, [91](#)  
DeviceStreamChannelCount, [91](#)  
DeviceStreamChannelEndianness, [91](#)  
DeviceStreamChannelLink, [92](#)  
DeviceStreamChannelPacketSize, [92](#)  
DeviceStreamChannelSelector, [92](#)  
DeviceStreamChannelType, [92](#)  
DeviceTapGeometry, [92](#)  
DeviceTemperature, [92](#)  
DeviceTemperatureSelector, [92](#)  
DeviceTLType, [92](#)  
DeviceTLVersionMajor, [93](#)  
DeviceTLVersionMinor, [93](#)  
DeviceTLVersionSubMinor, [93](#)  
DeviceType, [93](#)  
DeviceUptime, [93](#)  
DeviceUserID, [93](#)  
DeviceVendorName, [93](#)  
DeviceVersion, [93](#)  
EncoderDivider, [94](#)  
EncoderMode, [94](#)  
EncoderOutputMode, [94](#)  
EncoderReset, [94](#)  
EncoderResetActivation, [94](#)  
EncoderResetSource, [94](#)  
EncoderSelector, [94](#)  
EncoderSourceA, [94](#)  
EncoderSourceB, [95](#)  
EncoderStatus, [95](#)  
EncoderTimeout, [95](#)  
EncoderValue, [95](#)  
EncoderValueAtReset, [95](#)  
EnumerationCount, [95](#)  
EventAcquisitionEnd, [95](#)  
EventAcquisitionEndFrameID, [95](#)  
EventAcquisitionEndTimestamp, [96](#)  
EventAcquisitionError, [96](#)  
EventAcquisitionErrorFrameID, [96](#)  
EventAcquisitionErrorTimestamp, [96](#)  
EventAcquisitionStart, [96](#)  
EventAcquisitionStartFrameID, [96](#)  
EventAcquisitionStartTimestamp, [96](#)  
EventAcquisitionTransferEnd, [96](#)  
EventAcquisitionTransferEndFrameID, [97](#)  
EventAcquisitionTransferEndTimestamp, [97](#)  
EventAcquisitionTransferStart, [97](#)  
EventAcquisitionTransferStartFrameID, [97](#)  
EventAcquisitionTransferStartTimestamp, [97](#)  
EventAcquisitionTrigger, [97](#)  
EventAcquisitionTriggerFrameID, [97](#)  
EventAcquisitionTriggerTimestamp, [97](#)  
EventActionLate, [98](#)  
EventActionLateFrameID, [98](#)  
EventActionLateTimestamp, [98](#)  
EventCounter0End, [98](#)  
EventCounter0EndFrameID, [98](#)  
EventCounter0EndTimestamp, [98](#)  
EventCounter0Start, [98](#)  
EventCounter0StartFrameID, [98](#)  
EventCounter0StartTimestamp, [99](#)  
EventCounter1End, [99](#)  
EventCounter1EndFrameID, [99](#)  
EventCounter1EndTimestamp, [99](#)  
EventCounter1Start, [99](#)  
EventCounter1StartFrameID, [99](#)  
EventCounter1StartTimestamp, [99](#)  
EventEncoder0Restarted, [99](#)  
EventEncoder0RestartedFrameID, [100](#)  
EventEncoder0RestartedTimestamp, [100](#)  
EventEncoder0Stopped, [100](#)  
EventEncoder0StoppedFrameID, [100](#)  
EventEncoder0StoppedTimestamp, [100](#)  
EventEncoder1Restarted, [100](#)  
EventEncoder1RestartedFrameID, [100](#)  
EventEncoder1RestartedTimestamp, [100](#)  
EventEncoder1Stopped, [101](#)  
EventEncoder1StoppedFrameID, [101](#)  
EventEncoder1StoppedTimestamp, [101](#)  
EventError, [101](#)  
EventErrorCode, [101](#)

- EventErrorFrameID, 101
- EventErrorTimestamp, 101
- EventExposureEnd, 101
- EventExposureEndFrameID, 102
- EventExposureEndTimestamp, 102
- EventExposureStart, 102
- EventExposureStartFrameID, 102
- EventExposureStartTimestamp, 102
- EventFrameBurstEnd, 102
- EventFrameBurstEndFrameID, 102
- EventFrameBurstEndTimestamp, 102
- EventFrameBurstStart, 103
- EventFrameBurstStartFrameID, 103
- EventFrameBurstStartTimestamp, 103
- EventFrameEnd, 103
- EventFrameEndFrameID, 103
- EventFrameEndTimestamp, 103
- EventFrameStart, 103
- EventFrameStartFrameID, 103
- EventFrameStartTimestamp, 104
- EventFrameTransferEnd, 104
- EventFrameTransferEndFrameID, 104
- EventFrameTransferEndTimestamp, 104
- EventFrameTransferStart, 104
- EventFrameTransferStartFrameID, 104
- EventFrameTransferStartTimestamp, 104
- EventFrameTrigger, 104
- EventFrameTriggerFrameID, 105
- EventFrameTriggerTimestamp, 105
- EventLine0AnyEdge, 105
- EventLine0AnyEdgeFrameID, 105
- EventLine0AnyEdgeTimestamp, 105
- EventLine0FallingEdge, 105
- EventLine0FallingEdgeFrameID, 105
- EventLine0FallingEdgeTimestamp, 105
- EventLine0RisingEdge, 106
- EventLine0RisingEdgeFrameID, 106
- EventLine0RisingEdgeTimestamp, 106
- EventLine1AnyEdge, 106
- EventLine1AnyEdgeFrameID, 106
- EventLine1AnyEdgeTimestamp, 106
- EventLine1FallingEdge, 106
- EventLine1FallingEdgeFrameID, 106
- EventLine1FallingEdgeTimestamp, 107
- EventLine1RisingEdge, 107
- EventLine1RisingEdgeFrameID, 107
- EventLine1RisingEdgeTimestamp, 107
- EventLinkSpeedChange, 107
- EventLinkSpeedChangeFrameID, 107
- EventLinkSpeedChangeTimestamp, 107
- EventLinkTrigger0, 107
- EventLinkTrigger0FrameID, 108
- EventLinkTrigger0Timestamp, 108
- EventLinkTrigger1, 108
- EventLinkTrigger1FrameID, 108
- EventLinkTrigger1Timestamp, 108
- EventNotification, 108
- EventSelector, 108
- EventSequencerSetChange, 108
- EventSequencerSetChangeFrameID, 109
- EventSequencerSetChangeTimestamp, 109
- EventSerialData, 109
- EventSerialDataLength, 109
- EventSerialPortReceive, 109
- EventSerialPortReceiveTimestamp, 109
- EventSerialReceiveOverflow, 109
- EventStream0TransferBlockEnd, 109
- EventStream0TransferBlockEndFrameID, 110
- EventStream0TransferBlockEndTimestamp, 110
- EventStream0TransferBlockStart, 110
- EventStream0TransferBlockStartFrameID, 110
- EventStream0TransferBlockStartTimestamp, 110
- EventStream0TransferBlockTrigger, 110
- EventStream0TransferBlockTriggerFrameID, 110
- EventStream0TransferBlockTriggerTimestamp, 110
- EventStream0TransferBurstEnd, 111
- EventStream0TransferBurstEndFrameID, 111
- EventStream0TransferBurstEndTimestamp, 111
- EventStream0TransferBurstStart, 111
- EventStream0TransferBurstStartFrameID, 111
- EventStream0TransferBurstStartTimestamp, 111
- EventStream0TransferEnd, 111
- EventStream0TransferEndFrameID, 111
- EventStream0TransferEndTimestamp, 112
- EventStream0TransferOverflow, 112
- EventStream0TransferOverflowFrameID, 112
- EventStream0TransferOverflowTimestamp, 112
- EventStream0TransferPause, 112
- EventStream0TransferPauseFrameID, 112
- EventStream0TransferPauseTimestamp, 112
- EventStream0TransferResume, 112
- EventStream0TransferResumeFrameID, 113
- EventStream0TransferResumeTimestamp, 113
- EventStream0TransferStart, 113
- EventStream0TransferStartFrameID, 113
- EventStream0TransferStartTimestamp, 113
- EventTest, 113
- EventTestTimestamp, 113
- EventTimer0End, 113
- EventTimer0EndFrameID, 114
- EventTimer0EndTimestamp, 114
- EventTimer0Start, 114
- EventTimer0StartFrameID, 114
- EventTimer0StartTimestamp, 114
- EventTimer1End, 114
- EventTimer1EndFrameID, 114
- EventTimer1EndTimestamp, 114
- EventTimer1Start, 115
- EventTimer1StartFrameID, 115
- EventTimer1StartTimestamp, 115
- ExposureActiveMode, 115
- ExposureAuto, 115
- ExposureMode, 115
- ExposureTime, 115
- ExposureTimeMode, 115
- ExposureTimeSelector, 116

- FactoryReset, [116](#)
- FileAccessBuffer, [116](#)
- FileAccessLength, [116](#)
- FileAccessOffset, [116](#)
- FileOpenMode, [116](#)
- FileOperationExecute, [116](#)
- FileOperationResult, [116](#)
- FileOperationSelector, [117](#)
- FileOperationStatus, [117](#)
- FileSelector, [117](#)
- FileSize, [117](#)
- Gain, [117](#)
- GainAuto, [117](#)
- GainAutoBalance, [117](#)
- GainSelector, [117](#)
- Gamma, [118](#)
- GammaEnable, [118](#)
- GevActiveLinkCount, [118](#)
- GevCCP, [118](#)
- GevCurrentDefaultGateway, [118](#)
- GevCurrentIPAddress, [118](#)
- GevCurrentIPConfigurationDHCP, [118](#)
- GevCurrentIPConfigurationLLA, [118](#)
- GevCurrentIPConfigurationPersistentIP, [119](#)
- GevCurrentPhysicalLinkConfiguration, [119](#)
- GevCurrentSubnetMask, [119](#)
- GevDiscoveryAckDelay, [119](#)
- GevFirstURL, [119](#)
- GevGVCPExtendedStatusCodes, [119](#)
- GevGVCPExtendedStatusCodesSelector, [119](#)
- GevGVCPHeartbeatDisable, [119](#)
- GevGVCPPendingAck, [120](#)
- GevGVCPPendingTimeout, [120](#)
- GevGVSPExtendedIDMode, [120](#)
- GevHeartbeatTimeout, [120](#)
- GevIEEE1588, [120](#)
- GevIEEE1588ClockAccuracy, [120](#)
- GevIEEE1588Mode, [120](#)
- GevIEEE1588Status, [120](#)
- GevInterfaceSelector, [121](#)
- GevIPConfigurationStatus, [121](#)
- GevMACAddress, [121](#)
- GevMCDA, [121](#)
- GevMCPHostPort, [121](#)
- GevMCRC, [121](#)
- GevMCSP, [121](#)
- GevMCTT, [121](#)
- GevNumberOfInterfaces, [122](#)
- GevPAUSEFrameReception, [122](#)
- GevPAUSEFrameTransmission, [122](#)
- GevPersistentDefaultGateway, [122](#)
- GevPersistentIPAddress, [122](#)
- GevPersistentSubnetMask, [122](#)
- GevPhysicalLinkConfiguration, [122](#)
- GevPrimaryApplicationIPAddress, [122](#)
- GevPrimaryApplicationSocket, [123](#)
- GevPrimaryApplicationSwitchoverKey, [123](#)
- GevSCCFGAllInTransmission, [123](#)
- GevSCCFGExtendedChunkData, [123](#)
- GevSCCFGPacketResendDestination, [123](#)
- GevSCCFGUnconditionalStreaming, [123](#)
- GevSCDA, [123](#)
- GevSCPD, [123](#)
- GevSCPDDirection, [124](#)
- GevSCPHostPort, [124](#)
- GevSCPInterfaceIndex, [124](#)
- GevSCPSBigEndian, [124](#)
- GevSCPSDoNotFragment, [124](#)
- GevSCPSFireTestPacket, [124](#)
- GevSCPSPacketSize, [124](#)
- GevSCSP, [124](#)
- GevSCZoneConfigurationLock, [125](#)
- GevSCZoneCount, [125](#)
- GevSCZoneDirectionAll, [125](#)
- GevSecondURL, [125](#)
- GevStreamChannelSelector, [125](#)
- GevSupportedOption, [125](#)
- GevSupportedOptionSelector, [125](#)
- GevTimestampTickFrequency, [125](#)
- GuiXmlManifestAddress, [126](#)
- Height, [126](#)
- HeightMax, [126](#)
- ImageComponentEnable, [126](#)
- ImageComponentSelector, [126](#)
- ImageCompressionBitrate, [126](#)
- ImageCompressionJPEGFormatOption, [126](#)
- ImageCompressionMode, [126](#)
- ImageCompressionQuality, [127](#)
- ImageCompressionRateOption, [127](#)
- IspEnable, [127](#)
- LineFilterWidth, [127](#)
- LineFormat, [127](#)
- LineInputFilterSelector, [127](#)
- LineInverter, [127](#)
- LineMode, [127](#)
- LinePitch, [128](#)
- LineSelector, [128](#)
- LineSource, [128](#)
- LineStatus, [128](#)
- LineStatusAll, [128](#)
- LinkErrorCount, [128](#)
- LinkUptime, [128](#)
- LogicBlockLUTInputActivation, [128](#)
- LogicBlockLUTInputSelector, [129](#)
- LogicBlockLUTInputSource, [129](#)
- LogicBlockLUTOutputValue, [129](#)
- LogicBlockLUTOutputValueAll, [129](#)
- LogicBlockLUTRowIndex, [129](#)
- LogicBlockLUTSelector, [129](#)
- LogicBlockSelector, [129](#)
- LUTEnable, [129](#)
- LUTIndex, [130](#)
- LUTSelector, [130](#)
- LUTValue, [130](#)
- LUTValueAll, [130](#)
- MaxDeviceResetTime, [130](#)

- OffsetX, [130](#)
- OffsetY, [130](#)
- PacketResendRequestCount, [130](#)
- PayloadSize, [131](#)
- PixelColorFilter, [131](#)
- PixelDynamicRangeMax, [131](#)
- PixelDynamicRangeMin, [131](#)
- PixelFormat, [131](#)
- PixelFormatInfoID, [131](#)
- PixelFormatInfoSelector, [131](#)
- PixelSize, [131](#)
- PowerSupplyCurrent, [132](#)
- PowerSupplyVoltage, [132](#)
- RegionDestination, [132](#)
- RegionMode, [132](#)
- RegionSelector, [132](#)
- ReverseX, [132](#)
- ReverseY, [132](#)
- RgbTransformLightSource, [132](#)
- Saturation, [133](#)
- SaturationEnable, [133](#)
- Scan3dAxisMax, [133](#)
- Scan3dAxisMin, [133](#)
- Scan3dCoordinateOffset, [133](#)
- Scan3dCoordinateReferenceSelector, [133](#)
- Scan3dCoordinateReferenceValue, [133](#)
- Scan3dCoordinateScale, [133](#)
- Scan3dCoordinateSelector, [134](#)
- Scan3dCoordinateSystem, [134](#)
- Scan3dCoordinateSystemReference, [134](#)
- Scan3dCoordinateTransformSelector, [134](#)
- Scan3dDistanceUnit, [134](#)
- Scan3dInvalidDataFlag, [134](#)
- Scan3dInvalidDataValue, [134](#)
- Scan3dOutputMode, [134](#)
- Scan3dTransformValue, [135](#)
- SensorDescription, [135](#)
- SensorDigitizationTaps, [135](#)
- SensorHeight, [135](#)
- SensorShutterMode, [135](#)
- SensorTaps, [135](#)
- SensorWidth, [135](#)
- SequencerConfigurationMode, [135](#)
- SequencerConfigurationValid, [136](#)
- SequencerFeatureEnable, [136](#)
- SequencerMode, [136](#)
- SequencerPathSelector, [136](#)
- SequencerSetActive, [136](#)
- SequencerSetLoad, [136](#)
- SequencerSetNext, [136](#)
- SequencerSetSave, [136](#)
- SequencerSetSelector, [137](#)
- SequencerSetStart, [137](#)
- SequencerSetValid, [137](#)
- SequencerTriggerActivation, [137](#)
- SequencerTriggerSource, [137](#)
- SerialPortBaudRate, [137](#)
- SerialPortDataBits, [137](#)
- SerialPortParity, [137](#)
- SerialPortSelector, [138](#)
- SerialPortSource, [138](#)
- SerialPortStopBits, [138](#)
- SerialReceiveFramingErrorCount, [138](#)
- SerialReceiveParityErrorCount, [138](#)
- SerialReceiveQueueClear, [138](#)
- SerialReceiveQueueCurrentCharacterCount, [138](#)
- SerialReceiveQueueMaxCharacterCount, [138](#)
- SerialTransmitQueueCurrentCharacterCount, [139](#)
- SerialTransmitQueueMaxCharacterCount, [139](#)
- Sharpening, [139](#)
- SharpeningAuto, [139](#)
- SharpeningEnable, [139](#)
- SharpeningThreshold, [139](#)
- SoftwareSignalPulse, [139](#)
- SoftwareSignalSelector, [139](#)
- SourceCount, [140](#)
- SourceSelector, [140](#)
- Test0001, [140](#)
- TestEventGenerate, [140](#)
- TestPattern, [140](#)
- TestPatternGeneratorSelector, [140](#)
- TestPendingAck, [140](#)
- TimerDelay, [140](#)
- TimerDuration, [141](#)
- TimerReset, [141](#)
- TimerSelector, [141](#)
- TimerStatus, [141](#)
- TimerTriggerActivation, [141](#)
- TimerTriggerSource, [141](#)
- TimerValue, [141](#)
- Timestamp, [141](#)
- TimestampLatch, [142](#)
- TimestampLatchValue, [142](#)
- TimestampReset, [142](#)
- TLParamsLocked, [142](#)
- TransferAbort, [142](#)
- TransferBlockCount, [142](#)
- TransferBurstCount, [142](#)
- TransferComponentSelector, [142](#)
- TransferControlMode, [143](#)
- TransferOperationMode, [143](#)
- TransferPause, [143](#)
- TransferQueueCurrentBlockCount, [143](#)
- TransferQueueMaxBlockCount, [143](#)
- TransferQueueMode, [143](#)
- TransferQueueOverflowCount, [143](#)
- TransferResume, [143](#)
- TransferSelector, [144](#)
- TransferStart, [144](#)
- TransferStatus, [144](#)
- TransferStatusSelector, [144](#)
- TransferStop, [144](#)
- TransferStreamChannel, [144](#)
- TransferTriggerActivation, [144](#)
- TransferTriggerMode, [144](#)
- TransferTriggerSelector, [145](#)

- TransferTriggerSource, [145](#)
- TriggerActivation, [145](#)
- TriggerDelay, [145](#)
- TriggerDivider, [145](#)
- TriggerEventTest, [145](#)
- TriggerMode, [145](#)
- TriggerMultiplier, [145](#)
- TriggerOverlap, [146](#)
- TriggerSelector, [146](#)
- TriggerSoftware, [146](#)
- TriggerSource, [146](#)
- UserOutputSelector, [146](#)
- UserOutputValue, [146](#)
- UserOutputValueAll, [146](#)
- UserOutputValueAllMask, [146](#)
- UserSetDefault, [147](#)
- UserSetFeatureEnable, [147](#)
- UserSetLoad, [147](#)
- UserSetSave, [147](#)
- UserSetSelector, [147](#)
- V3\_3Enable, [147](#)
- WhiteClip, [147](#)
- WhiteClipSelector, [147](#)
- Width, [148](#)
- WidthMax, [148](#)
- \_quickSpinTLDevice, [148](#)
  - DeviceAccessStatus, [149](#)
  - DeviceCurrentSpeed, [149](#)
  - DeviceDisplayName, [149](#)
  - DeviceDriverVersion, [149](#)
  - DeviceEndianessMechanism, [149](#)
  - DeviceID, [150](#)
  - DeviceInstanceId, [150](#)
  - DevicesUpdater, [150](#)
  - DeviceLinkSpeed, [150](#)
  - DeviceLocation, [150](#)
  - DeviceModelName, [150](#)
  - DeviceMulticastMonitorMode, [150](#)
  - DeviceSerialNumber, [150](#)
  - DeviceType, [151](#)
  - DeviceU3VProtocol, [151](#)
  - DeviceUserID, [151](#)
  - DeviceVendorName, [151](#)
  - DeviceVersion, [151](#)
  - GenICamXMLLocation, [151](#)
  - GenICamXMLPath, [151](#)
  - GevCCP, [151](#)
  - GevDeviceAutoForceIP, [152](#)
  - GevDeviceDiscoverMaximumPacketSize, [152](#)
  - GevDeviceForceGateway, [152](#)
  - GevDeviceForceIP, [152](#)
  - GevDeviceForceIPAddress, [152](#)
  - GevDeviceForceSubnetMask, [152](#)
  - GevDeviceGateway, [152](#)
  - GevDeviceIPAddress, [152](#)
  - GevDevicesWrongSubnet, [153](#)
  - GevDeviceMACAddress, [153](#)
  - GevDeviceMaximumPacketSize, [153](#)
  - GevDeviceMaximumRetryCount, [153](#)
  - GevDeviceModelsBigEndian, [153](#)
  - GevDevicePort, [153](#)
  - GevDeviceReadAndWriteTimeout, [153](#)
  - GevDeviceSubnetMask, [153](#)
  - GevVersionMajor, [154](#)
  - GevVersionMinor, [154](#)
  - GUIXMLLocation, [154](#)
  - GUIXMLPath, [154](#)
- \_quickSpinTLInterface, [154](#)
  - ActionCommand, [155](#)
  - DeviceAccessStatus, [155](#)
  - DeviceCount, [155](#)
  - DeviceID, [156](#)
  - DeviceModelName, [156](#)
  - DeviceSelector, [156](#)
  - DeviceSerialNumber, [156](#)
  - DeviceUnlock, [156](#)
  - DeviceUpdateList, [156](#)
  - DeviceVendorName, [156](#)
  - FilterDriverStatus, [156](#)
  - GevActionDeviceKey, [157](#)
  - GevActionGroupKey, [157](#)
  - GevActionGroupMask, [157](#)
  - GevActionTime, [157](#)
  - GevDeviceAutoForceIP, [157](#)
  - GevDeviceForceGateway, [157](#)
  - GevDeviceForceIP, [157](#)
  - GevDeviceForceIPAddress, [157](#)
  - GevDeviceForceSubnetMask, [158](#)
  - GevDeviceGateway, [158](#)
  - GevDeviceIPAddress, [158](#)
  - GevDeviceMACAddress, [158](#)
  - GevDeviceSubnetMask, [158](#)
  - GevInterfaceGateway, [158](#)
  - GevInterfaceGatewaySelector, [158](#)
  - GevInterfaceMACAddress, [158](#)
  - GevInterfaceMTU, [159](#)
  - GevInterfaceReceiveLinkSpeed, [159](#)
  - GevInterfaceSubnetIPAddress, [159](#)
  - GevInterfaceSubnetMask, [159](#)
  - GevInterfaceSubnetSelector, [159](#)
  - GevInterfaceTransmitLinkSpeed, [159](#)
  - HostAdapterDriverVersion, [159](#)
  - HostAdapterName, [159](#)
  - HostAdapterVendor, [160](#)
  - IncompatibleDeviceCount, [160](#)
  - IncompatibleDeviceID, [160](#)
  - IncompatibleDeviceModelName, [160](#)
  - IncompatibleDeviceSelector, [160](#)
  - IncompatibleDeviceVendorName, [160](#)
  - IncompatibleGevDeviceIPAddress, [160](#)
  - IncompatibleGevDeviceMACAddress, [160](#)
  - IncompatibleGevDeviceSubnetMask, [161](#)
  - InterfaceDisplayName, [161](#)
  - InterfaceID, [161](#)
  - InterfaceType, [161](#)
  - POEStatus, [161](#)

- [\\_quickSpinTLStream, 162](#)
  - [GevFailedPacketCount, 162](#)
  - [GevMaximumNumberResendRequests, 162](#)
  - [GevPacketResendMode, 162](#)
  - [GevPacketResendTimeout, 163](#)
  - [GevResendPacketCount, 163](#)
  - [GevResendRequestCount, 163](#)
  - [GevTotalPacketCount, 163](#)
  - [StreamAnnounceBufferMinimum, 163](#)
  - [StreamAnnouncedBufferCount, 163](#)
  - [StreamBlockTransferSize, 163](#)
  - [StreamBufferAlignment, 163](#)
  - [StreamBufferCountManual, 164](#)
  - [StreamBufferCountMax, 164](#)
  - [StreamBufferCountMode, 164](#)
  - [StreamBufferCountResult, 164](#)
  - [StreamBufferHandlingMode, 164](#)
  - [StreamChunkCountMaximum, 164](#)
  - [StreamCRCCheckEnable, 164](#)
  - [StreamDeliveredFrameCount, 164](#)
  - [StreamFailedBufferCount, 165](#)
  - [StreamID, 165](#)
  - [StreamInputBufferCount, 165](#)
  - [StreamIsGrabbing, 165](#)
  - [StreamLostFrameCount, 165](#)
  - [StreamOutputBufferCount, 165](#)
  - [StreamStartedFrameCount, 165](#)
  - [StreamType, 165](#)
- [\\_quickSpinTLSystem, 166](#)
  - [EnumerateGEVInterfaces, 166](#)
  - [GenTLSFNCVersionMajor, 166](#)
  - [GenTLSFNCVersionMinor, 167](#)
  - [GenTLVersionSubMinor, 167](#)
  - [GenTLVersionMajor, 167](#)
  - [GenTLVersionMinor, 167](#)
  - [GevInterfaceDefaultGateway, 167](#)
  - [GevInterfaceDefaultIPAddress, 167](#)
  - [GevInterfaceDefaultSubnetMask, 167](#)
  - [GevInterfaceMACAddress, 167](#)
  - [GevVersionMajor, 168](#)
  - [GevVersionMinor, 168](#)
  - [InterfaceDisplayName, 168](#)
  - [InterfaceID, 168](#)
  - [InterfaceSelector, 168](#)
  - [InterfaceUpdateList, 168](#)
  - [TLDisplayName, 168](#)
  - [TLFileName, 168](#)
  - [TLID, 169](#)
  - [TLModelName, 169](#)
  - [TLPath, 169](#)
  - [TLType, 169](#)
  - [TLVendorName, 169](#)
  - [TLVersion, 169](#)
- [\\_spinAVIOption, 170](#)
  - [frameRate, 170](#)
  - [reserved, 170](#)
- [\\_spinAccessMode](#)
  - [SpinnakerGenApiDefsC.h, 457](#)
- [\\_spinAcquisitionModeEnums](#)
  - [CameraDefsC.h, 219](#)
- [\\_spinAcquisitionStatusSelectorEnums](#)
  - [CameraDefsC.h, 220](#)
- [\\_spinActionUnconditionalModeEnums](#)
  - [CameraDefsC.h, 220](#)
- [\\_spinAdcBitDepthEnums](#)
  - [CameraDefsC.h, 220](#)
- [\\_spinAutoAlgorithmSelectorEnums](#)
  - [CameraDefsC.h, 221](#)
- [\\_spinAutoExposureControlPriorityEnums](#)
  - [CameraDefsC.h, 221](#)
- [\\_spinAutoExposureLightingModeEnums](#)
  - [CameraDefsC.h, 221](#)
- [\\_spinAutoExposureMeteringModeEnums](#)
  - [CameraDefsC.h, 222](#)
- [\\_spinAutoExposureTargetGreyValueAutoEnums](#)
  - [CameraDefsC.h, 222](#)
- [\\_spinBMPOption, 170](#)
  - [indexedColor\\_8bit, 171](#)
  - [reserved, 171](#)
- [\\_spinBalanceRatioSelectorEnums](#)
  - [CameraDefsC.h, 223](#)
- [\\_spinBalanceWhiteAutoEnums](#)
  - [CameraDefsC.h, 223](#)
- [\\_spinBalanceWhiteAutoProfileEnums](#)
  - [CameraDefsC.h, 223](#)
- [\\_spinBinningHorizontalModeEnums](#)
  - [CameraDefsC.h, 224](#)
- [\\_spinBinningSelectorEnums](#)
  - [CameraDefsC.h, 224](#)
- [\\_spinBinningVerticalModeEnums](#)
  - [CameraDefsC.h, 224](#)
- [\\_spinBlackLevelAutoBalanceEnums](#)
  - [CameraDefsC.h, 225](#)
- [\\_spinBlackLevelAutoEnums](#)
  - [CameraDefsC.h, 225](#)
- [\\_spinBlackLevelSelectorEnums](#)
  - [CameraDefsC.h, 225](#)
- [\\_spinCachingMode](#)
  - [SpinnakerGenApiDefsC.h, 457](#)
- [\\_spinChunkBlackLevelSelectorEnums](#)
  - [CameraDefsC.h, 226](#)
- [\\_spinChunkCounterSelectorEnums](#)
  - [CameraDefsC.h, 226](#)
- [\\_spinChunkData, 171](#)
  - [m\\_blackLevel, 172](#)
  - [m\\_counterValue, 172](#)
  - [m\\_cRC, 173](#)
  - [m\\_encoderValue, 173](#)
  - [m\\_exposureEndLineStatusAll, 173](#)
  - [m\\_exposureTime, 173](#)
  - [m\\_frameID, 173](#)
  - [m\\_gain, 173](#)
  - [m\\_height, 173](#)
  - [m\\_image, 173](#)
  - [m\\_inferenceConfidence, 174](#)
  - [m\\_inferenceFrameID, 174](#)



- m\_inferenceResult, [174](#)
- m\_linePitch, [174](#)
- m\_lineStatusAll, [174](#)
- m\_offsetX, [174](#)
- m\_offsetY, [174](#)
- m\_partSelector, [174](#)
- m\_pixelDynamicRangeMax, [175](#)
- m\_pixelDynamicRangeMin, [175](#)
- m\_scan3dAxisMax, [175](#)
- m\_scan3dAxisMin, [175](#)
- m\_scan3dCoordinateOffset, [175](#)
- m\_scan3dCoordinateReferenceValue, [175](#)
- m\_scan3dCoordinateScale, [175](#)
- m\_scan3dInvalidDataValue, [175](#)
- m\_scan3dTransformValue, [176](#)
- m\_scanLineSelector, [176](#)
- m\_sequencerSetActive, [176](#)
- m\_serialDataLength, [176](#)
- m\_streamChannelID, [176](#)
- m\_timerValue, [176](#)
- m\_timestamp, [176](#)
- m\_timestampLatchValue, [176](#)
- m\_transferBlockID, [177](#)
- m\_transferQueueCurrentBlockCount, [177](#)
- m\_width, [177](#)
- \_spinChunkEncoderSelectorEnums
  - CameraDefsC.h, [226](#)
- \_spinChunkEncoderStatusEnums
  - CameraDefsC.h, [226](#)
- \_spinChunkExposureTimeSelectorEnums
  - CameraDefsC.h, [227](#)
- \_spinChunkGainSelectorEnums
  - CameraDefsC.h, [227](#)
- \_spinChunkImageComponentEnums
  - CameraDefsC.h, [228](#)
- \_spinChunkPixelFormatEnums
  - CameraDefsC.h, [228](#)
- \_spinChunkRegionIDEnums
  - CameraDefsC.h, [228](#)
- \_spinChunkScan3dCoordinateReferenceSelectorEnums
  - CameraDefsC.h, [229](#)
- \_spinChunkScan3dCoordinateSelectorEnums
  - CameraDefsC.h, [229](#)
- \_spinChunkScan3dCoordinateSystemEnums
  - CameraDefsC.h, [229](#)
- \_spinChunkScan3dCoordinateSystemReferenceEnums
  - CameraDefsC.h, [230](#)
- \_spinChunkScan3dCoordinateTransformSelectorEnums
  - CameraDefsC.h, [230](#)
- \_spinChunkScan3dDistanceUnitEnums
  - CameraDefsC.h, [230](#)
- \_spinChunkScan3dOutputModeEnums
  - CameraDefsC.h, [231](#)
- \_spinChunkSelectorEnums
  - CameraDefsC.h, [232](#)
- \_spinChunkSourceIDEnums
  - CameraDefsC.h, [232](#)
- \_spinChunkTimerSelectorEnums
  - CameraDefsC.h, [233](#)
- \_spinChunkTransferStreamIDEnums
  - CameraDefsC.h, [233](#)
- \_spinCIConfigurationEnums
  - CameraDefsC.h, [233](#)
- \_spinCITimeSlotsCountEnums
  - CameraDefsC.h, [234](#)
- \_spinColorProcessingAlgorithm
  - SpinnakerDefsC.h, [408](#)
- \_spinColorTransformationSelectorEnums
  - CameraDefsC.h, [234](#)
- \_spinColorTransformationValueSelectorEnums
  - CameraDefsC.h, [234](#)
- \_spinCounterEventActivationEnums
  - CameraDefsC.h, [235](#)
- \_spinCounterEventSourceEnums
  - CameraDefsC.h, [235](#)
- \_spinCounterResetActivationEnums
  - CameraDefsC.h, [236](#)
- \_spinCounterResetSourceEnums
  - CameraDefsC.h, [236](#)
- \_spinCounterSelectorEnums
  - CameraDefsC.h, [237](#)
- \_spinCounterStatusEnums
  - CameraDefsC.h, [237](#)
- \_spinCounterTriggerActivationEnums
  - CameraDefsC.h, [238](#)
- \_spinCounterTriggerSourceEnums
  - CameraDefsC.h, [238](#)
- \_spinCxpConnectionTestModeEnums
  - CameraDefsC.h, [239](#)
- \_spinCxpLinkConfigurationEnums
  - CameraDefsC.h, [239](#)
- \_spinCxpLinkConfigurationPreferredEnums
  - CameraDefsC.h, [240](#)
- \_spinCxpLinkConfigurationStatusEnums
  - CameraDefsC.h, [241](#)
- \_spinCxpPoCxpStatusEnums
  - CameraDefsC.h, [242](#)
- \_spinDecimationHorizontalModeEnums
  - CameraDefsC.h, [242](#)
- \_spinDecimationSelectorEnums
  - CameraDefsC.h, [242](#)
- \_spinDecimationVerticalModeEnums
  - CameraDefsC.h, [243](#)
- \_spinDefectCorrectionModeEnums
  - CameraDefsC.h, [243](#)
- \_spinDeinterlacingEnums
  - CameraDefsC.h, [243](#)
- \_spinDeviceCharacterSetEnums
  - CameraDefsC.h, [244](#)
- \_spinDeviceClockSelectorEnums
  - CameraDefsC.h, [244](#)
- \_spinDeviceConnectionStatusEnums
  - CameraDefsC.h, [244](#)
- \_spinDeviceIndicatorModeEnums
  - CameraDefsC.h, [245](#)
- \_spinDeviceLinkHeartbeatModeEnums

- CameraDefsC.h, [245](#)
- \_spinDeviceLinkThroughputLimitModeEnums
  - CameraDefsC.h, [245](#)
- \_spinDevicePowerSupplySelectorEnums
  - CameraDefsC.h, [245](#)
- \_spinDeviceRegistersEndiannessEnums
  - CameraDefsC.h, [246](#)
- \_spinDeviceScanTypeEnums
  - CameraDefsC.h, [246](#)
- \_spinDeviceSerialPortBaudRateEnums
  - CameraDefsC.h, [246](#)
- \_spinDeviceSerialPortSelectorEnums
  - CameraDefsC.h, [247](#)
- \_spinDeviceStreamChannelEndiannessEnums
  - CameraDefsC.h, [247](#)
- \_spinDeviceStreamChannelTypeEnums
  - CameraDefsC.h, [247](#)
- \_spinDeviceTLTypeEnums
  - CameraDefsC.h, [249](#)
- \_spinDeviceTapGeometryEnums
  - CameraDefsC.h, [248](#)
- \_spinDeviceTemperatureSelectorEnums
  - CameraDefsC.h, [249](#)
- \_spinDeviceTypeEnums
  - CameraDefsC.h, [250](#)
- \_spinDisplayNotation
  - SpinnakerGenApiDefsC.h, [457](#)
- \_spinEncoderModeEnums
  - CameraDefsC.h, [250](#)
- \_spinEncoderOutputModeEnums
  - CameraDefsC.h, [250](#)
- \_spinEncoderResetActivationEnums
  - CameraDefsC.h, [251](#)
- \_spinEncoderResetSourceEnums
  - CameraDefsC.h, [251](#)
- \_spinEncoderSelectorEnums
  - CameraDefsC.h, [252](#)
- \_spinEncoderSourceAEnums
  - CameraDefsC.h, [253](#)
- \_spinEncoderSourceBEnums
  - CameraDefsC.h, [253](#)
- \_spinEncoderStatusEnums
  - CameraDefsC.h, [253](#)
- \_spinEndianness
  - SpinnakerGenApiDefsC.h, [458](#)
- \_spinError
  - SpinnakerDefsC.h, [409](#)
- \_spinEventNotificationEnums
  - CameraDefsC.h, [254](#)
- \_spinEventSelectorEnums
  - CameraDefsC.h, [254](#)
- \_spinExposureActiveModeEnums
  - CameraDefsC.h, [254](#)
- \_spinExposureAutoEnums
  - CameraDefsC.h, [254](#)
- \_spinExposureModeEnums
  - CameraDefsC.h, [255](#)
- \_spinExposureTimeModeEnums
  - CameraDefsC.h, [255](#)
- \_spinExposureTimeSelectorEnums
  - CameraDefsC.h, [256](#)
- \_spinFileOpenModeEnums
  - CameraDefsC.h, [256](#)
- \_spinFileOperationSelectorEnums
  - CameraDefsC.h, [256](#)
- \_spinFileOperationStatusEnums
  - CameraDefsC.h, [257](#)
- \_spinFileSelectorEnums
  - CameraDefsC.h, [257](#)
- \_spinGainAutoBalanceEnums
  - CameraDefsC.h, [257](#)
- \_spinGainAutoEnums
  - CameraDefsC.h, [259](#)
- \_spinGainSelectorEnums
  - CameraDefsC.h, [259](#)
- \_spinGevCCPEnums
  - CameraDefsC.h, [259](#)
- \_spinGevCurrentPhysicalLinkConfigurationEnums
  - CameraDefsC.h, [260](#)
- \_spinGevGVCPExtendedStatusCodesSelectorEnums
  - CameraDefsC.h, [260](#)
- \_spinGevGVSPExtendedIDModeEnums
  - CameraDefsC.h, [260](#)
- \_spinGevIEEE1588ClockAccuracyEnums
  - CameraDefsC.h, [261](#)
- \_spinGevIEEE1588ModeEnums
  - CameraDefsC.h, [261](#)
- \_spinGevIEEE1588StatusEnums
  - CameraDefsC.h, [261](#)
- \_spinGevIPConfigurationStatusEnums
  - CameraDefsC.h, [262](#)
- \_spinGevPhysicalLinkConfigurationEnums
  - CameraDefsC.h, [262](#)
- \_spinGevSupportedOptionSelectorEnums
  - CameraDefsC.h, [262](#)
- \_spinH264Option, [177](#)
  - bitrate, [178](#)
  - frameRate, [178](#)
  - height, [178](#)
  - reserved, [178](#)
  - width, [178](#)
- \_spinImageComponentSelectorEnums
  - CameraDefsC.h, [263](#)
- \_spinImageCompressionJPEGFormatOptionEnums
  - CameraDefsC.h, [264](#)
- \_spinImageCompressionModeEnums
  - CameraDefsC.h, [264](#)
- \_spinImageCompressionRateOptionEnums
  - CameraDefsC.h, [265](#)
- \_spinImageFileFormat
  - SpinnakerDefsC.h, [410](#)
- \_spinImageStatus
  - SpinnakerDefsC.h, [411](#)
- \_spinIncMode
  - SpinnakerGenApiDefsC.h, [458](#)
- \_spinInputDirection



- SpinnakerGenApiDefsC.h, [458](#)
- \_spinInterfaceType
  - SpinnakerGenApiDefsC.h, [459](#)
- \_spinJPEGOption, [179](#)
  - progressive, [179](#)
  - quality, [179](#)
  - reserved, [179](#)
- \_spinJPG2Option, [180](#)
  - quality, [180](#)
  - reserved, [180](#)
- \_spinLUTSelectorEnums
  - CameraDefsC.h, [269](#)
- \_spinLibraryVersion, [181](#)
  - build, [181](#)
  - major, [181](#)
  - minor, [181](#)
  - type, [181](#)
- \_spinLineFormatEnums
  - CameraDefsC.h, [265](#)
- \_spinLineInputFilterSelectorEnums
  - CameraDefsC.h, [265](#)
- \_spinLineModeEnums
  - CameraDefsC.h, [266](#)
- \_spinLineSelectorEnums
  - CameraDefsC.h, [266](#)
- \_spinLineSourceEnums
  - CameraDefsC.h, [266](#)
- \_spinLinkType
  - SpinnakerGenApiDefsC.h, [460](#)
- \_spinLogLevel
  - SpinnakerDefsC.h, [411](#)
- \_spinLogicBlockLUTInputActivationEnums
  - CameraDefsC.h, [267](#)
- \_spinLogicBlockLUTInputSelectorEnums
  - CameraDefsC.h, [267](#)
- \_spinLogicBlockLUTInputSourceEnums
  - CameraDefsC.h, [268](#)
- \_spinLogicBlockLUTSelectorEnums
  - CameraDefsC.h, [268](#)
- \_spinLogicBlockSelectorEnums
  - CameraDefsC.h, [269](#)
- \_spinMJPGOption, [182](#)
  - frameRate, [182](#)
  - quality, [182](#)
  - reserved, [182](#)
- \_spinNameSpace
  - SpinnakerGenApiDefsC.h, [460](#)
- \_spinNodeType
  - SpinnakerGenApiDefsC.h, [461](#)
- \_spinPGMOption, [183](#)
  - binaryFile, [183](#)
  - reserved, [183](#)
- \_spinPNGOption, [184](#)
  - compressionLevel, [184](#)
  - interlaced, [184](#)
  - reserved, [184](#)
- \_spinPPMOption, [185](#)
  - binaryFile, [185](#)
  - reserved, [185](#)
- \_spinPayloadTypeInfoFoldS
  - SpinnakerDefsC.h, [412](#)
- \_spinPixelColorFilterEnums
  - CameraDefsC.h, [269](#)
- \_spinPixelFormatEnums
  - CameraDefsC.h, [270](#)
- \_spinPixelFormatInfoSelectorEnums
  - CameraDefsC.h, [275](#)
- \_spinPixelFormatNamespaceID
  - SpinnakerDefsC.h, [412](#)
- \_spinPixelSizeEnums
  - CameraDefsC.h, [281](#)
- \_spinRegionDestinationEnums
  - CameraDefsC.h, [282](#)
- \_spinRegionModeEnums
  - CameraDefsC.h, [282](#)
- \_spinRegionSelectorEnums
  - CameraDefsC.h, [282](#)
- \_spinRepresentation
  - SpinnakerGenApiDefsC.h, [461](#)
- \_spinRgbTransformLightSourceEnums
  - CameraDefsC.h, [283](#)
- \_spinScan3dCoordinateReferenceSelectorEnums
  - CameraDefsC.h, [283](#)
- \_spinScan3dCoordinateSelectorEnums
  - CameraDefsC.h, [284](#)
- \_spinScan3dCoordinateSystemEnums
  - CameraDefsC.h, [284](#)
- \_spinScan3dCoordinateSystemReferenceEnums
  - CameraDefsC.h, [284](#)
- \_spinScan3dCoordinateTransformSelectorEnums
  - CameraDefsC.h, [285](#)
- \_spinScan3dDistanceUnitEnums
  - CameraDefsC.h, [285](#)
- \_spinScan3dOutputModeEnums
  - CameraDefsC.h, [285](#)
- \_spinSensorDigitizationTapsEnums
  - CameraDefsC.h, [286](#)
- \_spinSensorShutterModeEnums
  - CameraDefsC.h, [287](#)
- \_spinSensorTapsEnums
  - CameraDefsC.h, [287](#)
- \_spinSequencerConfigurationModeEnums
  - CameraDefsC.h, [287](#)
- \_spinSequencerConfigurationValidEnums
  - CameraDefsC.h, [288](#)
- \_spinSequencerModeEnums
  - CameraDefsC.h, [288](#)
- \_spinSequencerSetValidEnums
  - CameraDefsC.h, [288](#)
- \_spinSequencerTriggerActivationEnums
  - CameraDefsC.h, [289](#)
- \_spinSequencerTriggerSourceEnums
  - CameraDefsC.h, [289](#)
- \_spinSerialPortBaudRateEnums
  - CameraDefsC.h, [289](#)
- \_spinSerialPortParityEnums

- CameraDefsC.h, [290](#)
- \_spinSerialPortSelectorEnums
  - CameraDefsC.h, [290](#)
- \_spinSerialPortSourceEnums
  - CameraDefsC.h, [291](#)
- \_spinSerialPortStopBitsEnums
  - CameraDefsC.h, [291](#)
- \_spinSign
  - SpinnakerGenApiDefsC.h, [461](#)
- \_spinSlope
  - SpinnakerGenApiDefsC.h, [462](#)
- \_spinSoftwareSignalSelectorEnums
  - CameraDefsC.h, [291](#)
- \_spinSourceSelectorEnums
  - CameraDefsC.h, [292](#)
- \_spinStandardNameSpace
  - SpinnakerGenApiDefsC.h, [462](#)
- \_spinStatisticsChannel
  - SpinnakerDefsC.h, [413](#)
- \_spinTIFFOption, [185](#)
  - compression, [186](#)
  - reserved, [186](#)
- \_spinTLDeviceAccessStatusEnums
  - TransportLayerDefsC.h, [469](#)
- \_spinTLDeviceCurrentSpeedEnums
  - TransportLayerDefsC.h, [469](#)
- \_spinTLDeviceEndianessMechanismEnums
  - TransportLayerDefsC.h, [469](#)
- \_spinTLDeviceTypeEnums
  - TransportLayerDefsC.h, [471](#)
- \_spinTLFilterDriverStatusEnums
  - TransportLayerDefsC.h, [471](#)
- \_spinTLGUIXMLLocationEnums
  - TransportLayerDefsC.h, [472](#)
- \_spinTLGenICamXMLLocationEnums
  - TransportLayerDefsC.h, [471](#)
- \_spinTLGevCCPEnums
  - TransportLayerDefsC.h, [472](#)
- \_spinTLInterfaceTypeEnums
  - TransportLayerDefsC.h, [472](#)
- \_spinTLPOEStatusEnums
  - TransportLayerDefsC.h, [473](#)
- \_spinTLStreamBufferCountModeEnums
  - TransportLayerDefsC.h, [473](#)
- \_spinTLStreamBufferHandlingModeEnums
  - TransportLayerDefsC.h, [473](#)
- \_spinTLStreamTypeEnums
  - TransportLayerDefsC.h, [474](#)
- \_spinTLTLTypeEnums
  - TransportLayerDefsC.h, [474](#)
- \_spinTestPatternEnums
  - CameraDefsC.h, [292](#)
- \_spinTestPatternGeneratorSelectorEnums
  - CameraDefsC.h, [292](#)
- \_spinTimerSelectorEnums
  - CameraDefsC.h, [293](#)
- \_spinTimerStatusEnums
  - CameraDefsC.h, [293](#)
- \_spinTimerTriggerActivationEnums
  - CameraDefsC.h, [293](#)
- \_spinTimerTriggerSourceEnums
  - CameraDefsC.h, [294](#)
- \_spinTransferComponentSelectorEnums
  - CameraDefsC.h, [295](#)
- \_spinTransferControlModeEnums
  - CameraDefsC.h, [295](#)
- \_spinTransferOperationModeEnums
  - CameraDefsC.h, [296](#)
- \_spinTransferQueueModeEnums
  - CameraDefsC.h, [296](#)
- \_spinTransferSelectorEnums
  - CameraDefsC.h, [296](#)
- \_spinTransferStatusSelectorEnums
  - CameraDefsC.h, [297](#)
- \_spinTransferTriggerActivationEnums
  - CameraDefsC.h, [297](#)
- \_spinTransferTriggerModeEnums
  - CameraDefsC.h, [297](#)
- \_spinTransferTriggerSelectorEnums
  - CameraDefsC.h, [298](#)
- \_spinTransferTriggerSourceEnums
  - CameraDefsC.h, [298](#)
- \_spinTriggerActivationEnums
  - CameraDefsC.h, [299](#)
- \_spinTriggerModeEnums
  - CameraDefsC.h, [300](#)
- \_spinTriggerOverlapEnums
  - CameraDefsC.h, [300](#)
- \_spinTriggerSelectorEnums
  - CameraDefsC.h, [300](#)
- \_spinTriggerSourceEnums
  - CameraDefsC.h, [300](#)
- \_spinUserOutputSelectorEnums
  - CameraDefsC.h, [301](#)
- \_spinUserSetDefaultEnums
  - CameraDefsC.h, [301](#)
- \_spinUserSetSelectorEnums
  - CameraDefsC.h, [302](#)
- \_spinVisibility
  - SpinnakerGenApiDefsC.h, [463](#)
- \_spinWhiteClipSelectorEnums
  - CameraDefsC.h, [302](#)
- \_spinXMLValidation
  - SpinnakerGenApiDefsC.h, [463](#)
- \_spinYesNo
  - SpinnakerGenApiDefsC.h, [463](#)
- AasRoiEnable
  - \_quickSpin, [64](#)
- AasRoiHeight
  - \_quickSpin, [64](#)
- AasRoiOffsetX
  - \_quickSpin, [64](#)
- AasRoiOffsetY
  - \_quickSpin, [64](#)
- AasRoiWidth
  - \_quickSpin, [64](#)

- AcquisitionAbort
  - [\\_quickSpin](#), 65
- AcquisitionArm
  - [\\_quickSpin](#), 65
- AcquisitionBurstFrameCount
  - [\\_quickSpin](#), 65
- AcquisitionFrameCount
  - [\\_quickSpin](#), 65
- AcquisitionFrameRate
  - [\\_quickSpin](#), 65
- AcquisitionFrameRateEnable
  - [\\_quickSpin](#), 65
- AcquisitionLineRate
  - [\\_quickSpin](#), 65
- AcquisitionMode
  - [\\_quickSpin](#), 65
- AcquisitionMode\_Continuous
  - [CameraDefsC.h](#), 220
- AcquisitionMode\_MultiFrame
  - [CameraDefsC.h](#), 220
- AcquisitionMode\_SingleFrame
  - [CameraDefsC.h](#), 220
- AcquisitionResultingFrameRate
  - [\\_quickSpin](#), 66
- AcquisitionStart
  - [\\_quickSpin](#), 66
- AcquisitionStatus
  - [\\_quickSpin](#), 66
- AcquisitionStatusSelector
  - [\\_quickSpin](#), 66
- AcquisitionStatusSelector\_AcquisitionActive
  - [CameraDefsC.h](#), 220
- AcquisitionStatusSelector\_AcquisitionTransfer
  - [CameraDefsC.h](#), 220
- AcquisitionStatusSelector\_AcquisitionTriggerWait
  - [CameraDefsC.h](#), 220
- AcquisitionStatusSelector\_ExposureActive
  - [CameraDefsC.h](#), 220
- AcquisitionStatusSelector\_FrameActive
  - [CameraDefsC.h](#), 220
- AcquisitionStatusSelector\_FrameTriggerWait
  - [CameraDefsC.h](#), 220
- AcquisitionStop
  - [\\_quickSpin](#), 66
- ACTION\_COMMAND\_STATUS\_ACTION\_LATE
  - [SpinnakerDefsC.h](#), 408
- ACTION\_COMMAND\_STATUS\_ERROR
  - [SpinnakerDefsC.h](#), 408
- ACTION\_COMMAND\_STATUS\_NO\_REF\_TIME
  - [SpinnakerDefsC.h](#), 408
- ACTION\_COMMAND\_STATUS\_OK
  - [SpinnakerDefsC.h](#), 408
- ACTION\_COMMAND\_STATUS\_OVERFLOW
  - [SpinnakerDefsC.h](#), 408
- ActionCommand
  - [\\_quickSpinTLInterface](#), 155
- ActionDeviceKey
  - [\\_quickSpin](#), 66
- ActionGroupKey
  - [\\_quickSpin](#), 66
- ActionGroupMask
  - [\\_quickSpin](#), 66
- ActionQueueSize
  - [\\_quickSpin](#), 67
- ActionSelector
  - [\\_quickSpin](#), 67
- ActionUnconditionalMode
  - [\\_quickSpin](#), 67
- ActionUnconditionalMode\_Off
  - [CameraDefsC.h](#), 220
- ActionUnconditionalMode\_On
  - [CameraDefsC.h](#), 220
- AdaptiveCompressionEnable
  - [\\_quickSpin](#), 67
- AdcBitDepth
  - [\\_quickSpin](#), 67
- AdcBitDepth\_Bit10
  - [CameraDefsC.h](#), 221
- AdcBitDepth\_Bit12
  - [CameraDefsC.h](#), 221
- AdcBitDepth\_Bit14
  - [CameraDefsC.h](#), 221
- AdcBitDepth\_Bit8
  - [CameraDefsC.h](#), 221
- ADOBE\_DEFLATE
  - [SpinnakerDefsC.h](#), 413
- aPAUSEMACCtrlFramesReceived
  - [\\_quickSpin](#), 67
- aPAUSEMACCtrlFramesTransmitted
  - [\\_quickSpin](#), 67
- AutoAlgorithmSelector
  - [\\_quickSpin](#), 67
- AutoAlgorithmSelector\_Ae
  - [CameraDefsC.h](#), 221
- AutoAlgorithmSelector\_Awb
  - [CameraDefsC.h](#), 221
- AutoExposureControlLoopDamping
  - [\\_quickSpin](#), 68
- AutoExposureControlPriority
  - [\\_quickSpin](#), 68
- AutoExposureControlPriority\_ExposureTime
  - [CameraDefsC.h](#), 221
- AutoExposureControlPriority\_Gain
  - [CameraDefsC.h](#), 221
- AutoExposureEVCompensation
  - [\\_quickSpin](#), 68
- AutoExposureExposureTimeLowerLimit
  - [\\_quickSpin](#), 68
- AutoExposureExposureTimeUpperLimit
  - [\\_quickSpin](#), 68
- AutoExposureGainLowerLimit
  - [\\_quickSpin](#), 68
- AutoExposureGainUpperLimit
  - [\\_quickSpin](#), 68
- AutoExposureGreyValueLowerLimit
  - [\\_quickSpin](#), 68

- AutoExposureGreyValueUpperLimit
  - \_quickSpin, [69](#)
- AutoExposureLightingMode
  - \_quickSpin, [69](#)
- AutoExposureLightingMode\_AutoDetect
  - CameraDefsC.h, [222](#)
- AutoExposureLightingMode\_Backlight
  - CameraDefsC.h, [222](#)
- AutoExposureLightingMode\_Frontlight
  - CameraDefsC.h, [222](#)
- AutoExposureLightingMode\_Normal
  - CameraDefsC.h, [222](#)
- AutoExposureMeteringMode
  - \_quickSpin, [69](#)
- AutoExposureMeteringMode\_Average
  - CameraDefsC.h, [222](#)
- AutoExposureMeteringMode\_CenterWeighted
  - CameraDefsC.h, [222](#)
- AutoExposureMeteringMode\_HistogramPeak
  - CameraDefsC.h, [222](#)
- AutoExposureMeteringMode\_Partial
  - CameraDefsC.h, [222](#)
- AutoExposureMeteringMode\_Spot
  - CameraDefsC.h, [222](#)
- AutoExposureTargetGreyValue
  - \_quickSpin, [69](#)
- AutoExposureTargetGreyValueAuto
  - \_quickSpin, [69](#)
- AutoExposureTargetGreyValueAuto\_Continuous
  - CameraDefsC.h, [222](#)
- AutoExposureTargetGreyValueAuto\_Off
  - CameraDefsC.h, [222](#)
- Automatic
  - SpinnakerGenApiDefsC.h, [462](#)
- BalanceRatio
  - \_quickSpin, [69](#)
- BalanceRatioSelector
  - \_quickSpin, [69](#)
- BalanceRatioSelector\_Blue
  - CameraDefsC.h, [223](#)
- BalanceRatioSelector\_Red
  - CameraDefsC.h, [223](#)
- BalanceWhiteAuto
  - \_quickSpin, [69](#)
- BalanceWhiteAuto\_Continuous
  - CameraDefsC.h, [223](#)
- BalanceWhiteAuto\_Off
  - CameraDefsC.h, [223](#)
- BalanceWhiteAuto\_Once
  - CameraDefsC.h, [223](#)
- BalanceWhiteAutoDamping
  - \_quickSpin, [70](#)
- BalanceWhiteAutoLowerLimit
  - \_quickSpin, [70](#)
- BalanceWhiteAutoProfile
  - \_quickSpin, [70](#)
- BalanceWhiteAutoProfile\_Indoor
  - CameraDefsC.h, [223](#)
- BalanceWhiteAutoProfile\_Outdoor
  - CameraDefsC.h, [223](#)
- BalanceWhiteAutoUpperLimit
  - \_quickSpin, [70](#)
- BaseNode
  - SpinnakerGenApiDefsC.h, [461](#)
- Beginner
  - SpinnakerGenApiDefsC.h, [463](#)
- BigEndian
  - SpinnakerGenApiDefsC.h, [458](#)
- BILINEAR
  - SpinnakerDefsC.h, [409](#)
- binaryFile
  - \_spinPGMOption, [183](#)
  - \_spinPPMOption, [185](#)
- BinningHorizontal
  - \_quickSpin, [70](#)
- BinningHorizontalMode
  - \_quickSpin, [70](#)
- BinningHorizontalMode\_Average
  - CameraDefsC.h, [224](#)
- BinningHorizontalMode\_Sum
  - CameraDefsC.h, [224](#)
- BinningSelector
  - \_quickSpin, [70](#)
- BinningSelector\_All
  - CameraDefsC.h, [224](#)
- BinningSelector\_ISP
  - CameraDefsC.h, [224](#)
- BinningSelector\_Sensor
  - CameraDefsC.h, [224](#)
- BinningVertical
  - \_quickSpin, [70](#)
- BinningVerticalMode
  - \_quickSpin, [71](#)
- BinningVerticalMode\_Average
  - CameraDefsC.h, [224](#)
- BinningVerticalMode\_Sum
  - CameraDefsC.h, [224](#)
- bitrate
  - \_spinH264Option, [178](#)
- BlackLevel
  - \_quickSpin, [71](#)
- BlackLevelAuto
  - \_quickSpin, [71](#)
- BlackLevelAuto\_Continuous
  - CameraDefsC.h, [225](#)
- BlackLevelAuto\_Off
  - CameraDefsC.h, [225](#)
- BlackLevelAuto\_Once
  - CameraDefsC.h, [225](#)
- BlackLevelAutoBalance
  - \_quickSpin, [71](#)
- BlackLevelAutoBalance\_Continuous
  - CameraDefsC.h, [225](#)
- BlackLevelAutoBalance\_Off
  - CameraDefsC.h, [225](#)
- BlackLevelAutoBalance\_Once

- CameraDefsC.h, [225](#)
- BlackLevelClampingEnable
  - \_quickSpin, [71](#)
- BlackLevelRaw
  - \_quickSpin, [71](#)
- BlackLevelSelector
  - \_quickSpin, [71](#)
- BlackLevelSelector\_All
  - CameraDefsC.h, [225](#)
- BlackLevelSelector\_Analog
  - CameraDefsC.h, [225](#)
- BlackLevelSelector\_Digital
  - CameraDefsC.h, [225](#)
- BLUE
  - SpinnakerDefsC.h, [413](#)
- BMP
  - SpinnakerDefsC.h, [410](#)
- bool8\_t
  - SpinnakerDefsC.h, [404](#)
- Boolean
  - SpinnakerGenApiDefsC.h, [461](#)
- BooleanNode
  - SpinnakerGenApiDefsC.h, [461](#)
- build
  - \_spinLibraryVersion, [181](#)
- Camera Access, [18](#)
- Camera Enumerations, [8](#)
- CameraDefsC.h
  - \_spinAcquisitionModeEnums, [219](#)
  - \_spinAcquisitionStatusSelectorEnums, [220](#)
  - \_spinActionUnconditionalModeEnums, [220](#)
  - \_spinAdcBitDepthEnums, [220](#)
  - \_spinAutoAlgorithmSelectorEnums, [221](#)
  - \_spinAutoExposureControlPriorityEnums, [221](#)
  - \_spinAutoExposureLightingModeEnums, [221](#)
  - \_spinAutoExposureMeteringModeEnums, [222](#)
  - \_spinAutoExposureTargetGreyValueAutoEnums, [222](#)
  - \_spinBalanceRatioSelectorEnums, [223](#)
  - \_spinBalanceWhiteAutoEnums, [223](#)
  - \_spinBalanceWhiteAutoProfileEnums, [223](#)
  - \_spinBinningHorizontalModeEnums, [224](#)
  - \_spinBinningSelectorEnums, [224](#)
  - \_spinBinningVerticalModeEnums, [224](#)
  - \_spinBlackLevelAutoBalanceEnums, [225](#)
  - \_spinBlackLevelAutoEnums, [225](#)
  - \_spinBlackLevelSelectorEnums, [225](#)
  - \_spinChunkBlackLevelSelectorEnums, [226](#)
  - \_spinChunkCounterSelectorEnums, [226](#)
  - \_spinChunkEncoderSelectorEnums, [226](#)
  - \_spinChunkEncoderStatusEnums, [226](#)
  - \_spinChunkExposureTimeSelectorEnums, [227](#)
  - \_spinChunkGainSelectorEnums, [227](#)
  - \_spinChunkImageComponentEnums, [228](#)
  - \_spinChunkPixelFormatEnums, [228](#)
  - \_spinChunkRegionIDEnums, [228](#)
  - \_spinChunkScan3dCoordinateReferenceSelectorEnums, [229](#)
  - \_spinChunkScan3dCoordinateSelectorEnums, [229](#)
  - \_spinChunkScan3dCoordinateSystemEnums, [229](#)
  - \_spinChunkScan3dCoordinateSystemReferenceEnums, [230](#)
  - \_spinChunkScan3dCoordinateTransformSelectorEnums, [230](#)
  - \_spinChunkScan3dDistanceUnitEnums, [230](#)
  - \_spinChunkScan3dOutputModeEnums, [231](#)
  - \_spinChunkSelectorEnums, [232](#)
  - \_spinChunkSourceIDEnums, [232](#)
  - \_spinChunkTimerSelectorEnums, [233](#)
  - \_spinChunkTransferStreamIDEnums, [233](#)
  - \_spinCIConfigurationEnums, [233](#)
  - \_spinCITimeSlotsCountEnums, [234](#)
  - \_spinColorTransformationSelectorEnums, [234](#)
  - \_spinColorTransformationValueSelectorEnums, [234](#)
  - \_spinCounterEventActivationEnums, [235](#)
  - \_spinCounterEventSourceEnums, [235](#)
  - \_spinCounterResetActivationEnums, [236](#)
  - \_spinCounterResetSourceEnums, [236](#)
  - \_spinCounterSelectorEnums, [237](#)
  - \_spinCounterStatusEnums, [237](#)
  - \_spinCounterTriggerActivationEnums, [238](#)
  - \_spinCounterTriggerSourceEnums, [238](#)
  - \_spinCxpConnectionTestModeEnums, [239](#)
  - \_spinCxpLinkConfigurationEnums, [239](#)
  - \_spinCxpLinkConfigurationPreferredEnums, [240](#)
  - \_spinCxpLinkConfigurationStatusEnums, [241](#)
  - \_spinCxpPoCxpStatusEnums, [242](#)
  - \_spinDecimationHorizontalModeEnums, [242](#)
  - \_spinDecimationSelectorEnums, [242](#)
  - \_spinDecimationVerticalModeEnums, [243](#)
  - \_spinDefectCorrectionModeEnums, [243](#)
  - \_spinDeinterlacingEnums, [243](#)
  - \_spinDeviceCharacterSetEnums, [244](#)
  - \_spinDeviceClockSelectorEnums, [244](#)
  - \_spinDeviceConnectionStatusEnums, [244](#)
  - \_spinDeviceIndicatorModeEnums, [245](#)
  - \_spinDeviceLinkHeartbeatModeEnums, [245](#)
  - \_spinDeviceLinkThroughputLimitModeEnums, [245](#)
  - \_spinDevicePowerSupplySelectorEnums, [245](#)
  - \_spinDeviceRegistersEndiannessEnums, [246](#)
  - \_spinDeviceScanTypeEnums, [246](#)
  - \_spinDeviceSerialPortBaudRateEnums, [246](#)
  - \_spinDeviceSerialPortSelectorEnums, [247](#)
  - \_spinDeviceStreamChannelEndiannessEnums, [247](#)
  - \_spinDeviceStreamChannelTypeEnums, [247](#)
  - \_spinDeviceTLTypeEnums, [249](#)
  - \_spinDeviceTapGeometryEnums, [248](#)
  - \_spinDeviceTemperatureSelectorEnums, [249](#)
  - \_spinDeviceTypeEnums, [250](#)
  - \_spinEncoderModeEnums, [250](#)
  - \_spinEncoderOutputModeEnums, [250](#)
  - \_spinEncoderResetActivationEnums, [251](#)
  - \_spinEncoderResetSourceEnums, [251](#)
  - \_spinEncoderSelectorEnums, [252](#)



- [\\_spinEncoderSourceAEnums, 253](#)
- [\\_spinEncoderSourceBEnums, 253](#)
- [\\_spinEncoderStatusEnums, 253](#)
- [\\_spinEventNotificationEnums, 254](#)
- [\\_spinEventSelectorEnums, 254](#)
- [\\_spinExposureActiveModeEnums, 254](#)
- [\\_spinExposureAutoEnums, 254](#)
- [\\_spinExposureModeEnums, 255](#)
- [\\_spinExposureTimeModeEnums, 255](#)
- [\\_spinExposureTimeSelectorEnums, 256](#)
- [\\_spinFileOpenModeEnums, 256](#)
- [\\_spinFileOperationSelectorEnums, 256](#)
- [\\_spinFileOperationStatusEnums, 257](#)
- [\\_spinFileSelectorEnums, 257](#)
- [\\_spinGainAutoBalanceEnums, 257](#)
- [\\_spinGainAutoEnums, 259](#)
- [\\_spinGainSelectorEnums, 259](#)
- [\\_spinGevCCPEnums, 259](#)
- [\\_spinGevCurrentPhysicalLinkConfigurationEnums, 260](#)
- [\\_spinGevGVCPExtendedStatusCodesSelectorEnums, 260](#)
- [\\_spinGevGVSPExtendedIDModeEnums, 260](#)
- [\\_spinGevIEEE1588ClockAccuracyEnums, 261](#)
- [\\_spinGevIEEE1588ModeEnums, 261](#)
- [\\_spinGevIEEE1588StatusEnums, 261](#)
- [\\_spinGevIPConfigurationStatusEnums, 262](#)
- [\\_spinGevPhysicalLinkConfigurationEnums, 262](#)
- [\\_spinGevSupportedOptionSelectorEnums, 262](#)
- [\\_spinImageComponentSelectorEnums, 263](#)
- [\\_spinImageCompressionJPEGFormatOptionEnums, 264](#)
- [\\_spinImageCompressionModeEnums, 264](#)
- [\\_spinImageCompressionRateOptionEnums, 265](#)
- [\\_spinLUTSelectorEnums, 269](#)
- [\\_spinLineFormatEnums, 265](#)
- [\\_spinLineInputFilterSelectorEnums, 265](#)
- [\\_spinLineModeEnums, 266](#)
- [\\_spinLineSelectorEnums, 266](#)
- [\\_spinLineSourceEnums, 266](#)
- [\\_spinLogicBlockLUTInputActivationEnums, 267](#)
- [\\_spinLogicBlockLUTInputSelectorEnums, 267](#)
- [\\_spinLogicBlockLUTInputSourceEnums, 268](#)
- [\\_spinLogicBlockLUTSelectorEnums, 268](#)
- [\\_spinLogicBlockSelectorEnums, 269](#)
- [\\_spinPixelColorFilterEnums, 269](#)
- [\\_spinPixelFormatEnums, 270](#)
- [\\_spinPixelFormatInfoSelectorEnums, 275](#)
- [\\_spinPixelSizeEnums, 281](#)
- [\\_spinRegionDestinationEnums, 282](#)
- [\\_spinRegionModeEnums, 282](#)
- [\\_spinRegionSelectorEnums, 282](#)
- [\\_spinRgbTransformLightSourceEnums, 283](#)
- [\\_spinScan3dCoordinateReferenceSelectorEnums, 283](#)
- [\\_spinScan3dCoordinateSelectorEnums, 284](#)
- [\\_spinScan3dCoordinateSystemEnums, 284](#)
- [\\_spinScan3dCoordinateSystemReferenceEnums, 284](#)
- [\\_spinScan3dCoordinateTransformSelectorEnums, 285](#)
- [\\_spinScan3dDistanceUnitEnums, 285](#)
- [\\_spinScan3dOutputModeEnums, 285](#)
- [\\_spinSensorDigitizationTapsEnums, 286](#)
- [\\_spinSensorShutterModeEnums, 287](#)
- [\\_spinSensorTapsEnums, 287](#)
- [\\_spinSequencerConfigurationModeEnums, 287](#)
- [\\_spinSequencerConfigurationValidEnums, 288](#)
- [\\_spinSequencerModeEnums, 288](#)
- [\\_spinSequencerSetValidEnums, 288](#)
- [\\_spinSequencerTriggerActivationEnums, 289](#)
- [\\_spinSequencerTriggerSourceEnums, 289](#)
- [\\_spinSerialPortBaudRateEnums, 289](#)
- [\\_spinSerialPortParityEnums, 290](#)
- [\\_spinSerialPortSelectorEnums, 290](#)
- [\\_spinSerialPortSourceEnums, 291](#)
- [\\_spinSerialPortStopBitsEnums, 291](#)
- [\\_spinSoftwareSignalSelectorEnums, 291](#)
- [\\_spinSourceSelectorEnums, 292](#)
- [\\_spinTestPatternEnums, 292](#)
- [\\_spinTestPatternGeneratorSelectorEnums, 292](#)
- [\\_spinTimerSelectorEnums, 293](#)
- [\\_spinTimerStatusEnums, 293](#)
- [\\_spinTimerTriggerActivationEnums, 293](#)
- [\\_spinTimerTriggerSourceEnums, 294](#)
- [\\_spinTransferComponentSelectorEnums, 295](#)
- [\\_spinTransferControlModeEnums, 295](#)
- [\\_spinTransferOperationModeEnums, 296](#)
- [\\_spinTransferQueueModeEnums, 296](#)
- [\\_spinTransferSelectorEnums, 296](#)
- [\\_spinTransferStatusSelectorEnums, 297](#)
- [\\_spinTransferTriggerActivationEnums, 297](#)
- [\\_spinTransferTriggerModeEnums, 297](#)
- [\\_spinTransferTriggerSelectorEnums, 298](#)
- [\\_spinTransferTriggerSourceEnums, 298](#)
- [\\_spinTriggerActivationEnums, 299](#)
- [\\_spinTriggerModeEnums, 300](#)
- [\\_spinTriggerOverlapEnums, 300](#)
- [\\_spinTriggerSelectorEnums, 300](#)
- [\\_spinTriggerSourceEnums, 300](#)
- [\\_spinUserOutputSelectorEnums, 301](#)
- [\\_spinUserSetDefaultEnums, 301](#)
- [\\_spinUserSetSelectorEnums, 302](#)
- [\\_spinWhiteClipSelectorEnums, 302](#)
- [AcquisitionMode\\_Continuous, 220](#)
- [AcquisitionMode\\_MultiFrame, 220](#)
- [AcquisitionMode\\_SingleFrame, 220](#)
- [AcquisitionStatusSelector\\_AcquisitionActive, 220](#)
- [AcquisitionStatusSelector\\_AcquisitionTransfer, 220](#)
- [AcquisitionStatusSelector\\_AcquisitionTriggerWait, 220](#)
- [AcquisitionStatusSelector\\_ExposureActive, 220](#)
- [AcquisitionStatusSelector\\_FrameActive, 220](#)
- [AcquisitionStatusSelector\\_FrameTriggerWait, 220](#)
- [ActionUnconditionalMode\\_Off, 220](#)

- ActionUnconditionalMode\_On, [220](#)
- AdcBitDepth\_Bit10, [221](#)
- AdcBitDepth\_Bit12, [221](#)
- AdcBitDepth\_Bit14, [221](#)
- AdcBitDepth\_Bit8, [221](#)
- AutoAlgorithmSelector\_Ae, [221](#)
- AutoAlgorithmSelector\_Awb, [221](#)
- AutoExposureControlPriority\_ExposureTime, [221](#)
- AutoExposureControlPriority\_Gain, [221](#)
- AutoExposureLightingMode\_AutoDetect, [222](#)
- AutoExposureLightingMode\_Backlight, [222](#)
- AutoExposureLightingMode\_Frontlight, [222](#)
- AutoExposureLightingMode\_Normal, [222](#)
- AutoExposureMeteringMode\_Average, [222](#)
- AutoExposureMeteringMode\_CenterWeighted, [222](#)
- AutoExposureMeteringMode\_HistogramPeak, [222](#)
- AutoExposureMeteringMode\_Partial, [222](#)
- AutoExposureMeteringMode\_Spot, [222](#)
- AutoExposureTargetGreyValueAuto\_Continuous, [222](#)
- AutoExposureTargetGreyValueAuto\_Off, [222](#)
- BalanceRatioSelector\_Blue, [223](#)
- BalanceRatioSelector\_Red, [223](#)
- BalanceWhiteAuto\_Continuous, [223](#)
- BalanceWhiteAuto\_Off, [223](#)
- BalanceWhiteAuto\_Once, [223](#)
- BalanceWhiteAutoProfile\_Indoor, [223](#)
- BalanceWhiteAutoProfile\_Outdoor, [223](#)
- BinningHorizontalMode\_Average, [224](#)
- BinningHorizontalMode\_Sum, [224](#)
- BinningSelector\_All, [224](#)
- BinningSelector\_ISP, [224](#)
- BinningSelector\_Sensor, [224](#)
- BinningVerticalMode\_Average, [224](#)
- BinningVerticalMode\_Sum, [224](#)
- BlackLevelAuto\_Continuous, [225](#)
- BlackLevelAuto\_Off, [225](#)
- BlackLevelAuto\_Once, [225](#)
- BlackLevelAutoBalance\_Continuous, [225](#)
- BlackLevelAutoBalance\_Off, [225](#)
- BlackLevelAutoBalance\_Once, [225](#)
- BlackLevelSelector\_All, [225](#)
- BlackLevelSelector\_Analog, [225](#)
- BlackLevelSelector\_Digital, [225](#)
- ChunkBlackLevelSelector\_All, [226](#)
- ChunkCounterSelector\_Counter0, [226](#)
- ChunkCounterSelector\_Counter1, [226](#)
- ChunkCounterSelector\_Counter2, [226](#)
- ChunkEncoderSelector\_Encoder0, [226](#)
- ChunkEncoderSelector\_Encoder1, [226](#)
- ChunkEncoderSelector\_Encoder2, [226](#)
- ChunkEncoderStatus\_EncoderDown, [227](#)
- ChunkEncoderStatus\_EncoderIdle, [227](#)
- ChunkEncoderStatus\_EncoderStatic, [227](#)
- ChunkEncoderStatus\_EncoderUp, [227](#)
- ChunkExposureTimeSelector\_Blue, [227](#)
- ChunkExposureTimeSelector\_Common, [227](#)
- ChunkExposureTimeSelector\_Cyan, [227](#)
- ChunkExposureTimeSelector\_Green, [227](#)
- ChunkExposureTimeSelector\_Infrared, [227](#)
- ChunkExposureTimeSelector\_Magenta, [227](#)
- ChunkExposureTimeSelector\_Red, [227](#)
- ChunkExposureTimeSelector\_Stage1, [227](#)
- ChunkExposureTimeSelector\_Stage2, [227](#)
- ChunkExposureTimeSelector\_Ultraviolet, [227](#)
- ChunkExposureTimeSelector\_Yellow, [227](#)
- ChunkGainSelector\_All, [227](#)
- ChunkGainSelector\_Blue, [227](#)
- ChunkGainSelector\_Green, [227](#)
- ChunkGainSelector\_Red, [227](#)
- ChunkImageComponent\_Color, [228](#)
- ChunkImageComponent\_Confidence, [228](#)
- ChunkImageComponent\_Disparity, [228](#)
- ChunkImageComponent\_Infrared, [228](#)
- ChunkImageComponent\_Intensity, [228](#)
- ChunkImageComponent\_Range, [228](#)
- ChunkImageComponent\_Scatter, [228](#)
- ChunkImageComponent\_Ultraviolet, [228](#)
- ChunkPixelFormat\_BayerBG8, [228](#)
- ChunkPixelFormat\_BayerGB8, [228](#)
- ChunkPixelFormat\_BayerGR8, [228](#)
- ChunkPixelFormat\_BayerRG8, [228](#)
- ChunkPixelFormat\_Mono12Packed, [228](#)
- ChunkPixelFormat\_Mono16, [228](#)
- ChunkPixelFormat\_Mono8, [228](#)
- ChunkPixelFormat\_RGB8Packed, [228](#)
- ChunkPixelFormat\_YCbCr601\_422\_8\_CbYCrY, [228](#)
- ChunkPixelFormat\_YUV422Packed, [228](#)
- ChunkRegionID\_Region0, [229](#)
- ChunkRegionID\_Region1, [229](#)
- ChunkRegionID\_Region2, [229](#)
- ChunkScan3dCoordinateReferenceSelector\_RotationX, [229](#)
- ChunkScan3dCoordinateReferenceSelector\_RotationY, [229](#)
- ChunkScan3dCoordinateReferenceSelector\_RotationZ, [229](#)
- ChunkScan3dCoordinateReferenceSelector\_TranslationX, [229](#)
- ChunkScan3dCoordinateReferenceSelector\_TranslationY, [229](#)
- ChunkScan3dCoordinateReferenceSelector\_TranslationZ, [229](#)
- ChunkScan3dCoordinateSelector\_CoordinateA, [229](#)
- ChunkScan3dCoordinateSelector\_CoordinateB, [229](#)
- ChunkScan3dCoordinateSelector\_CoordinateC, [229](#)
- ChunkScan3dCoordinateSystem\_Cartesian, [230](#)
- ChunkScan3dCoordinateSystem\_Cylindrical, [230](#)
- ChunkScan3dCoordinateSystem\_Spherical, [230](#)
- ChunkScan3dCoordinateSystemReference\_Anchor, [230](#)

- ChunkScan3dCoordinateSystemReference\_Transformed, [230](#)
- ChunkScan3dCoordinateTransformSelector\_RotationX, [230](#)
- ChunkScan3dCoordinateTransformSelector\_RotationY, [230](#)
- ChunkScan3dCoordinateTransformSelector\_RotationZ, [230](#)
- ChunkScan3dCoordinateTransformSelector\_TranslationX, [230](#)
- ChunkScan3dCoordinateTransformSelector\_TranslationY, [230](#)
- ChunkScan3dCoordinateTransformSelector\_TranslationZ, [230](#)
- ChunkScan3dDistanceUnit\_Inch, [231](#)
- ChunkScan3dDistanceUnit\_Millimeter, [231](#)
- ChunkScan3dOutputMode\_CalibratedABC\_Grid, [231](#)
- ChunkScan3dOutputMode\_CalibratedABC\_PointCloud, [231](#)
- ChunkScan3dOutputMode\_CalibratedAC, [231](#)
- ChunkScan3dOutputMode\_CalibratedAC\_Linescan, [231](#)
- ChunkScan3dOutputMode\_CalibratedC, [231](#)
- ChunkScan3dOutputMode\_CalibratedC\_Linescan, [231](#)
- ChunkScan3dOutputMode\_DisparityC, [232](#)
- ChunkScan3dOutputMode\_DisparityC\_Linescan, [232](#)
- ChunkScan3dOutputMode\_RectifiedC, [231](#)
- ChunkScan3dOutputMode\_RectifiedC\_Linescan, [232](#)
- ChunkScan3dOutputMode\_UncalibratedC, [231](#)
- ChunkSelector\_BlackLevel, [232](#)
- ChunkSelector\_CRC, [232](#)
- ChunkSelector\_ExposureEndLineStatusAll, [232](#)
- ChunkSelector\_ExposureTime, [232](#)
- ChunkSelector\_FrameID, [232](#)
- ChunkSelector\_Gain, [232](#)
- ChunkSelector\_Height, [232](#)
- ChunkSelector\_Image, [232](#)
- ChunkSelector\_OffsetX, [232](#)
- ChunkSelector\_OffsetY, [232](#)
- ChunkSelector\_PixelFormat, [232](#)
- ChunkSelector\_SequencerSetActive, [232](#)
- ChunkSelector\_SerialData, [232](#)
- ChunkSelector\_Timestamp, [232](#)
- ChunkSelector\_Width, [232](#)
- ChunkSourceID\_Source0, [233](#)
- ChunkSourceID\_Source1, [233](#)
- ChunkSourceID\_Source2, [233](#)
- ChunkTimerSelector\_Timer0, [233](#)
- ChunkTimerSelector\_Timer1, [233](#)
- ChunkTimerSelector\_Timer2, [233](#)
- ChunkTransferStreamID\_Stream0, [233](#)
- ChunkTransferStreamID\_Stream1, [233](#)
- ChunkTransferStreamID\_Stream2, [233](#)
- ChunkTransferStreamID\_Stream3, [233](#)
- CIConfiguration\_Base, [234](#)
- CIConfiguration\_DualBase, [234](#)
- CIConfiguration\_EightyBit, [234](#)
- CIConfiguration\_Full, [234](#)
- CIConfiguration\_Medium, [234](#)
- CITimeSlotsCount\_One, [234](#)
- CITimeSlotsCount\_Three, [234](#)
- CITimeSlotsCount\_Two, [234](#)
- ColorTransformationSelector\_RGBtoRGB, [234](#)
- ColorTransformationSelector\_RGBtoYUV, [234](#)
- ColorTransformationValueSelector\_Gain00, [235](#)
- ColorTransformationValueSelector\_Gain01, [235](#)
- ColorTransformationValueSelector\_Gain02, [235](#)
- ColorTransformationValueSelector\_Gain10, [235](#)
- ColorTransformationValueSelector\_Gain11, [235](#)
- ColorTransformationValueSelector\_Gain12, [235](#)
- ColorTransformationValueSelector\_Gain20, [235](#)
- ColorTransformationValueSelector\_Gain21, [235](#)
- ColorTransformationValueSelector\_Gain22, [235](#)
- ColorTransformationValueSelector\_Offset0, [235](#)
- ColorTransformationValueSelector\_Offset1, [235](#)
- ColorTransformationValueSelector\_Offset2, [235](#)
- CounterEventActivation\_AnyEdge, [235](#)
- CounterEventActivation\_FallingEdge, [235](#)
- CounterEventActivation\_LevelHigh, [235](#)
- CounterEventActivation\_LevelLow, [235](#)
- CounterEventActivation\_RisingEdge, [235](#)
- CounterEventSource\_Counter0End, [236](#)
- CounterEventSource\_Counter0Start, [236](#)
- CounterEventSource\_Counter1End, [236](#)
- CounterEventSource\_Counter1Start, [236](#)
- CounterEventSource\_ExposureEnd, [236](#)
- CounterEventSource\_ExposureStart, [236](#)
- CounterEventSource\_FrameTriggerWait, [236](#)
- CounterEventSource\_Line0, [235](#)
- CounterEventSource\_Line1, [236](#)
- CounterEventSource\_Line2, [236](#)
- CounterEventSource\_Line3, [236](#)
- CounterEventSource\_LogicBlock0, [236](#)
- CounterEventSource\_LogicBlock1, [236](#)
- CounterEventSource\_MHzTick, [235](#)
- CounterEventSource\_Off, [235](#)
- CounterEventSource\_UserOutput0, [236](#)
- CounterEventSource\_UserOutput1, [236](#)
- CounterEventSource\_UserOutput2, [236](#)
- CounterEventSource\_UserOutput3, [236](#)
- CounterResetActivation\_AnyEdge, [236](#)
- CounterResetActivation\_FallingEdge, [236](#)
- CounterResetActivation\_LevelHigh, [236](#)
- CounterResetActivation\_LevelLow, [236](#)
- CounterResetActivation\_RisingEdge, [236](#)
- CounterResetSource\_Counter0End, [237](#)
- CounterResetSource\_Counter0Start, [237](#)
- CounterResetSource\_Counter1End, [237](#)
- CounterResetSource\_Counter1Start, [237](#)
- CounterResetSource\_ExposureEnd, [237](#)
- CounterResetSource\_ExposureStart, [237](#)
- CounterResetSource\_FrameTriggerWait, [237](#)



CounterResetSource\_Line0, [237](#)  
CounterResetSource\_Line1, [237](#)  
CounterResetSource\_Line2, [237](#)  
CounterResetSource\_Line3, [237](#)  
CounterResetSource\_LogicBlock0, [237](#)  
CounterResetSource\_LogicBlock1, [237](#)  
CounterResetSource\_Off, [236](#)  
CounterResetSource\_UserOutput0, [237](#)  
CounterResetSource\_UserOutput1, [237](#)  
CounterResetSource\_UserOutput2, [237](#)  
CounterResetSource\_UserOutput3, [237](#)  
CounterSelector\_Counter0, [237](#)  
CounterSelector\_Counter1, [237](#)  
CounterStatus\_CounterActive, [237](#)  
CounterStatus\_CounterCompleted, [237](#)  
CounterStatus\_CounterIdle, [237](#)  
CounterStatus\_CounterOverflow, [237](#)  
CounterStatus\_CounterTriggerWait, [237](#)  
CounterTriggerActivation\_AnyEdge, [238](#)  
CounterTriggerActivation\_FallingEdge, [238](#)  
CounterTriggerActivation\_LevelHigh, [238](#)  
CounterTriggerActivation\_LevelLow, [238](#)  
CounterTriggerActivation\_RisingEdge, [238](#)  
CounterTriggerSource\_Counter0End, [238](#)  
CounterTriggerSource\_Counter0Start, [238](#)  
CounterTriggerSource\_Counter1End, [238](#)  
CounterTriggerSource\_Counter1Start, [238](#)  
CounterTriggerSource\_ExposureEnd, [238](#)  
CounterTriggerSource\_ExposureStart, [238](#)  
CounterTriggerSource\_FrameTriggerWait, [238](#)  
CounterTriggerSource\_Line0, [238](#)  
CounterTriggerSource\_Line1, [238](#)  
CounterTriggerSource\_Line2, [238](#)  
CounterTriggerSource\_Line3, [238](#)  
CounterTriggerSource\_LogicBlock0, [238](#)  
CounterTriggerSource\_LogicBlock1, [238](#)  
CounterTriggerSource\_Off, [238](#)  
CounterTriggerSource\_UserOutput0, [238](#)  
CounterTriggerSource\_UserOutput1, [238](#)  
CounterTriggerSource\_UserOutput2, [238](#)  
CounterTriggerSource\_UserOutput3, [238](#)  
CxpConnectionTestMode\_Mode1, [239](#)  
CxpConnectionTestMode\_Off, [239](#)  
CxpLinkConfiguration\_Auto, [239](#)  
CxpLinkConfiguration\_CXP1\_X1, [239](#)  
CxpLinkConfiguration\_CXP1\_X2, [239](#)  
CxpLinkConfiguration\_CXP1\_X3, [239](#)  
CxpLinkConfiguration\_CXP1\_X4, [239](#)  
CxpLinkConfiguration\_CXP1\_X5, [240](#)  
CxpLinkConfiguration\_CXP1\_X6, [240](#)  
CxpLinkConfiguration\_CXP2\_X1, [239](#)  
CxpLinkConfiguration\_CXP2\_X2, [239](#)  
CxpLinkConfiguration\_CXP2\_X3, [239](#)  
CxpLinkConfiguration\_CXP2\_X4, [239](#)  
CxpLinkConfiguration\_CXP2\_X5, [240](#)  
CxpLinkConfiguration\_CXP2\_X6, [240](#)  
CxpLinkConfiguration\_CXP3\_X1, [239](#)  
CxpLinkConfiguration\_CXP3\_X2, [239](#)  
CxpLinkConfiguration\_CXP3\_X3, [239](#)  
CxpLinkConfiguration\_CXP3\_X4, [239](#)  
CxpLinkConfiguration\_CXP3\_X5, [240](#)  
CxpLinkConfiguration\_CXP3\_X6, [240](#)  
CxpLinkConfiguration\_CXP5\_X1, [239](#)  
CxpLinkConfiguration\_CXP5\_X2, [239](#)  
CxpLinkConfiguration\_CXP5\_X3, [239](#)  
CxpLinkConfiguration\_CXP5\_X4, [239](#)  
CxpLinkConfiguration\_CXP5\_X5, [240](#)  
CxpLinkConfiguration\_CXP5\_X6, [240](#)  
CxpLinkConfiguration\_CXP6\_X1, [239](#)  
CxpLinkConfiguration\_CXP6\_X2, [239](#)  
CxpLinkConfiguration\_CXP6\_X3, [239](#)  
CxpLinkConfiguration\_CXP6\_X4, [239](#)  
CxpLinkConfiguration\_CXP6\_X5, [240](#)  
CxpLinkConfiguration\_CXP6\_X6, [240](#)  
CxpLinkConfigurationPreferred\_CXP1\_X1, [240](#)  
CxpLinkConfigurationPreferred\_CXP1\_X2, [240](#)  
CxpLinkConfigurationPreferred\_CXP1\_X3, [240](#)  
CxpLinkConfigurationPreferred\_CXP1\_X4, [240](#)  
CxpLinkConfigurationPreferred\_CXP1\_X5, [240](#)  
CxpLinkConfigurationPreferred\_CXP1\_X6, [241](#)  
CxpLinkConfigurationPreferred\_CXP2\_X1, [240](#)  
CxpLinkConfigurationPreferred\_CXP2\_X2, [240](#)  
CxpLinkConfigurationPreferred\_CXP2\_X3, [240](#)  
CxpLinkConfigurationPreferred\_CXP2\_X4, [240](#)  
CxpLinkConfigurationPreferred\_CXP2\_X5, [240](#)  
CxpLinkConfigurationPreferred\_CXP2\_X6, [241](#)  
CxpLinkConfigurationPreferred\_CXP3\_X1, [240](#)  
CxpLinkConfigurationPreferred\_CXP3\_X2, [240](#)  
CxpLinkConfigurationPreferred\_CXP3\_X3, [240](#)  
CxpLinkConfigurationPreferred\_CXP3\_X4, [240](#)  
CxpLinkConfigurationPreferred\_CXP3\_X5, [240](#)  
CxpLinkConfigurationPreferred\_CXP3\_X6, [241](#)  
CxpLinkConfigurationPreferred\_CXP5\_X1, [240](#)  
CxpLinkConfigurationPreferred\_CXP5\_X2, [240](#)  
CxpLinkConfigurationPreferred\_CXP5\_X3, [240](#)  
CxpLinkConfigurationPreferred\_CXP5\_X4, [240](#)  
CxpLinkConfigurationPreferred\_CXP5\_X5, [240](#)  
CxpLinkConfigurationPreferred\_CXP5\_X6, [241](#)  
CxpLinkConfigurationPreferred\_CXP6\_X1, [240](#)  
CxpLinkConfigurationPreferred\_CXP6\_X2, [240](#)  
CxpLinkConfigurationPreferred\_CXP6\_X3, [240](#)  
CxpLinkConfigurationPreferred\_CXP6\_X4, [240](#)  
CxpLinkConfigurationPreferred\_CXP6\_X5, [241](#)  
CxpLinkConfigurationPreferred\_CXP6\_X6, [241](#)  
CxpLinkConfigurationStatus\_CXP1\_X1, [241](#)  
CxpLinkConfigurationStatus\_CXP1\_X2, [241](#)  
CxpLinkConfigurationStatus\_CXP1\_X3, [241](#)  
CxpLinkConfigurationStatus\_CXP1\_X4, [241](#)  
CxpLinkConfigurationStatus\_CXP1\_X5, [241](#)  
CxpLinkConfigurationStatus\_CXP1\_X6, [242](#)  
CxpLinkConfigurationStatus\_CXP2\_X1, [241](#)  
CxpLinkConfigurationStatus\_CXP2\_X2, [241](#)  
CxpLinkConfigurationStatus\_CXP2\_X3, [241](#)  
CxpLinkConfigurationStatus\_CXP2\_X4, [241](#)  
CxpLinkConfigurationStatus\_CXP2\_X5, [241](#)  
CxpLinkConfigurationStatus\_CXP2\_X6, [242](#)

- CxpLinkConfigurationStatus\_CXP3\_X1, [241](#)
- CxpLinkConfigurationStatus\_CXP3\_X2, [241](#)
- CxpLinkConfigurationStatus\_CXP3\_X3, [241](#)
- CxpLinkConfigurationStatus\_CXP3\_X4, [241](#)
- CxpLinkConfigurationStatus\_CXP3\_X5, [241](#)
- CxpLinkConfigurationStatus\_CXP3\_X6, [242](#)
- CxpLinkConfigurationStatus\_CXP5\_X1, [241](#)
- CxpLinkConfigurationStatus\_CXP5\_X2, [241](#)
- CxpLinkConfigurationStatus\_CXP5\_X3, [241](#)
- CxpLinkConfigurationStatus\_CXP5\_X4, [241](#)
- CxpLinkConfigurationStatus\_CXP5\_X5, [241](#)
- CxpLinkConfigurationStatus\_CXP5\_X6, [242](#)
- CxpLinkConfigurationStatus\_CXP6\_X1, [241](#)
- CxpLinkConfigurationStatus\_CXP6\_X2, [241](#)
- CxpLinkConfigurationStatus\_CXP6\_X3, [241](#)
- CxpLinkConfigurationStatus\_CXP6\_X4, [241](#)
- CxpLinkConfigurationStatus\_CXP6\_X5, [242](#)
- CxpLinkConfigurationStatus\_CXP6\_X6, [242](#)
- CxpLinkConfigurationStatus\_None, [241](#)
- CxpLinkConfigurationStatus\_Pending, [241](#)
- CxpPoCxpStatus\_Auto, [242](#)
- CxpPoCxpStatus\_Off, [242](#)
- CxpPoCxpStatus\_Tripped, [242](#)
- DecimationHorizontalMode\_Discard, [242](#)
- DecimationSelector\_All, [243](#)
- DecimationSelector\_Sensor, [243](#)
- DecimationVerticalMode\_Discard, [243](#)
- DefectCorrectionMode\_Average, [243](#)
- DefectCorrectionMode\_Highlight, [243](#)
- DefectCorrectionMode\_Zero, [243](#)
- Deinterlacing\_LineDuplication, [244](#)
- Deinterlacing\_Off, [244](#)
- Deinterlacing\_Weave, [244](#)
- DeviceCharacterSet\_ASCII, [244](#)
- DeviceCharacterSet\_UTF8, [244](#)
- DeviceClockSelector\_CameraLink, [244](#)
- DeviceClockSelector\_Sensor, [244](#)
- DeviceClockSelector\_SensorDigitization, [244](#)
- DeviceConnectionStatus\_Active, [244](#)
- DeviceConnectionStatus\_Inactive, [244](#)
- DeviceIndicatorMode\_Active, [245](#)
- DeviceIndicatorMode\_ErrorStatus, [245](#)
- DeviceIndicatorMode\_Inactive, [245](#)
- DeviceLinkHeartbeatMode\_Off, [245](#)
- DeviceLinkHeartbeatMode\_On, [245](#)
- DeviceLinkThroughputLimitMode\_Off, [245](#)
- DeviceLinkThroughputLimitMode\_On, [245](#)
- DevicePowerSupplySelector\_External, [246](#)
- DeviceRegistersEndianness\_Big, [246](#)
- DeviceRegistersEndianness\_Little, [246](#)
- DeviceScanType\_Areascan, [246](#)
- DeviceSerialPortBaudRate\_Baud115200, [247](#)
- DeviceSerialPortBaudRate\_Baud19200, [247](#)
- DeviceSerialPortBaudRate\_Baud230400, [247](#)
- DeviceSerialPortBaudRate\_Baud38400, [247](#)
- DeviceSerialPortBaudRate\_Baud460800, [247](#)
- DeviceSerialPortBaudRate\_Baud57600, [247](#)
- DeviceSerialPortBaudRate\_Baud921600, [247](#)
- DeviceSerialPortBaudRate\_Baud9600, [247](#)
- DeviceSerialPortSelector\_CameraLink, [247](#)
- DeviceStreamChannelEndianness\_Big, [247](#)
- DeviceStreamChannelEndianness\_Little, [247](#)
- DeviceStreamChannelType\_Receiver, [248](#)
- DeviceStreamChannelType\_Transmitter, [248](#)
- DeviceTapGeometry\_Geometry\_10X, [249](#)
- DeviceTapGeometry\_Geometry\_10X\_1Y, [249](#)
- DeviceTapGeometry\_Geometry\_1X, [248](#)
- DeviceTapGeometry\_Geometry\_1X10, [249](#)
- DeviceTapGeometry\_Geometry\_1X10\_1Y, [249](#)
- DeviceTapGeometry\_Geometry\_1X2, [248](#)
- DeviceTapGeometry\_Geometry\_1X2\_1Y, [248](#)
- DeviceTapGeometry\_Geometry\_1X2\_1Y2, [248](#)
- DeviceTapGeometry\_Geometry\_1X2\_2YE, [248](#)
- DeviceTapGeometry\_Geometry\_1X3, [248](#)
- DeviceTapGeometry\_Geometry\_1X3\_1Y, [248](#)
- DeviceTapGeometry\_Geometry\_1X4, [248](#)
- DeviceTapGeometry\_Geometry\_1X4\_1Y, [248](#)
- DeviceTapGeometry\_Geometry\_1X8, [249](#)
- DeviceTapGeometry\_Geometry\_1X8\_1Y, [249](#)
- DeviceTapGeometry\_Geometry\_1X\_1Y, [248](#)
- DeviceTapGeometry\_Geometry\_1X\_1Y2, [248](#)
- DeviceTapGeometry\_Geometry\_1X\_2YE, [248](#)
- DeviceTapGeometry\_Geometry\_2X, [248](#)
- DeviceTapGeometry\_Geometry\_2X2, [248](#)
- DeviceTapGeometry\_Geometry\_2X2\_1Y, [248](#)
- DeviceTapGeometry\_Geometry\_2X2E, [248](#)
- DeviceTapGeometry\_Geometry\_2X2E\_1YGeometry\_2X2M\_1Y, [248](#)
- DeviceTapGeometry\_Geometry\_2X2E\_2YE, [249](#)
- DeviceTapGeometry\_Geometry\_2X2M, [249](#)
- DeviceTapGeometry\_Geometry\_2X\_1Y, [248](#)
- DeviceTapGeometry\_Geometry\_2X\_1Y2Geometry\_2XE\_1Y, [248](#)
- DeviceTapGeometry\_Geometry\_2X\_2YE, [248](#)
- DeviceTapGeometry\_Geometry\_2XE, [248](#)
- DeviceTapGeometry\_Geometry\_2XE\_1Y2, [248](#)
- DeviceTapGeometry\_Geometry\_2XE\_2YE, [248](#)
- DeviceTapGeometry\_Geometry\_2XM, [248](#)
- DeviceTapGeometry\_Geometry\_2XM\_1Y, [248](#)
- DeviceTapGeometry\_Geometry\_2XM\_1Y2, [248](#)
- DeviceTapGeometry\_Geometry\_2XM\_2YE, [248](#)
- DeviceTapGeometry\_Geometry\_3X, [248](#)
- DeviceTapGeometry\_Geometry\_3X\_1Y, [248](#)
- DeviceTapGeometry\_Geometry\_4X, [248](#)
- DeviceTapGeometry\_Geometry\_4X2, [249](#)
- DeviceTapGeometry\_Geometry\_4X2\_1Y, [249](#)
- DeviceTapGeometry\_Geometry\_4X2E, [249](#)
- DeviceTapGeometry\_Geometry\_4X2E\_1Y, [249](#)
- DeviceTapGeometry\_Geometry\_4X\_1Y, [248](#)
- DeviceTapGeometry\_Geometry\_8X, [249](#)
- DeviceTapGeometry\_Geometry\_8X\_1Y, [249](#)
- DeviceTemperatureSelector\_Sensor, [249](#)
- DeviceTLType\_CameraLink, [249](#)
- DeviceTLType\_CameraLinkHS, [249](#)
- DeviceTLType\_CoaXPress, [249](#)
- DeviceTLType\_Custom, [249](#)

- DeviceTLType\_GigEVision, [249](#)
- DeviceTLType\_USB3Vision, [249](#)
- DeviceType\_Peripheral, [250](#)
- DeviceType\_Receiver, [250](#)
- DeviceType\_Transceiver, [250](#)
- DeviceType\_Transmitter, [250](#)
- EncoderMode\_FourPhase, [250](#)
- EncoderMode\_HighResolution, [250](#)
- EncoderOutputMode\_DirectionDown, [251](#)
- EncoderOutputMode\_DirectionUp, [251](#)
- EncoderOutputMode\_Motion, [251](#)
- EncoderOutputMode\_Off, [250](#)
- EncoderOutputMode\_PositionDown, [251](#)
- EncoderOutputMode\_PositionUp, [250](#)
- EncoderResetActivation\_AnyEdge, [251](#)
- EncoderResetActivation\_FallingEdge, [251](#)
- EncoderResetActivation\_LevelHigh, [251](#)
- EncoderResetActivation\_LevelLow, [251](#)
- EncoderResetActivation\_RisingEdge, [251](#)
- EncoderResetSource\_AcquisitionEnd, [251](#)
- EncoderResetSource\_AcquisitionStart, [251](#)
- EncoderResetSource\_AcquisitionTrigger, [251](#)
- EncoderResetSource\_Action0, [252](#)
- EncoderResetSource\_Action1, [252](#)
- EncoderResetSource\_Action2, [252](#)
- EncoderResetSource\_Counter0End, [252](#)
- EncoderResetSource\_Counter0Start, [252](#)
- EncoderResetSource\_Counter1End, [252](#)
- EncoderResetSource\_Counter1Start, [252](#)
- EncoderResetSource\_Counter2End, [252](#)
- EncoderResetSource\_Counter2Start, [252](#)
- EncoderResetSource\_ExposureEnd, [252](#)
- EncoderResetSource\_ExposureStart, [251](#)
- EncoderResetSource\_FrameEnd, [251](#)
- EncoderResetSource\_FrameStart, [251](#)
- EncoderResetSource\_FrameTrigger, [251](#)
- EncoderResetSource\_Line0, [252](#)
- EncoderResetSource\_Line1, [252](#)
- EncoderResetSource\_Line2, [252](#)
- EncoderResetSource\_LinkTrigger0, [252](#)
- EncoderResetSource\_LinkTrigger1, [252](#)
- EncoderResetSource\_LinkTrigger2, [252](#)
- EncoderResetSource\_Off, [251](#)
- EncoderResetSource\_SoftwareSignal0, [252](#)
- EncoderResetSource\_SoftwareSignal1, [252](#)
- EncoderResetSource\_SoftwareSignal2, [252](#)
- EncoderResetSource\_Timer0End, [252](#)
- EncoderResetSource\_Timer0Start, [252](#)
- EncoderResetSource\_Timer1End, [252](#)
- EncoderResetSource\_Timer1Start, [252](#)
- EncoderResetSource\_Timer2End, [252](#)
- EncoderResetSource\_Timer2Start, [252](#)
- EncoderResetSource\_UserOutput0, [252](#)
- EncoderResetSource\_UserOutput1, [252](#)
- EncoderResetSource\_UserOutput2, [252](#)
- EncoderSelector\_Encoder0, [252](#)
- EncoderSelector\_Encoder1, [252](#)
- EncoderSelector\_Encoder2, [252](#)
- EncoderSourceA\_Line0, [253](#)
- EncoderSourceA\_Line1, [253](#)
- EncoderSourceA\_Line2, [253](#)
- EncoderSourceA\_Off, [253](#)
- EncoderSourceB\_Line0, [253](#)
- EncoderSourceB\_Line1, [253](#)
- EncoderSourceB\_Line2, [253](#)
- EncoderSourceB\_Off, [253](#)
- EncoderStatus\_EncoderDown, [253](#)
- EncoderStatus\_EncoderIdle, [253](#)
- EncoderStatus\_EncoderStatic, [253](#)
- EncoderStatus\_EncoderUp, [253](#)
- EventNotification\_Off, [254](#)
- EventNotification\_On, [254](#)
- EventSelector\_Error, [254](#)
- EventSelector\_ExposureEnd, [254](#)
- EventSelector\_SerialPortReceive, [254](#)
- ExposureActiveMode\_AllPixels, [254](#)
- ExposureActiveMode\_AnyPixels, [254](#)
- ExposureActiveMode\_Line1, [254](#)
- ExposureAuto\_Continuous, [255](#)
- ExposureAuto\_Off, [255](#)
- ExposureAuto\_Once, [255](#)
- ExposureMode\_Timed, [255](#)
- ExposureMode\_TriggerWidth, [255](#)
- ExposureTimeMode\_Common, [255](#)
- ExposureTimeMode\_Individual, [255](#)
- ExposureTimeSelector\_Blue, [256](#)
- ExposureTimeSelector\_Common, [256](#)
- ExposureTimeSelector\_Cyan, [256](#)
- ExposureTimeSelector\_Green, [256](#)
- ExposureTimeSelector\_Infrared, [256](#)
- ExposureTimeSelector\_Magenta, [256](#)
- ExposureTimeSelector\_Red, [256](#)
- ExposureTimeSelector\_Stage1, [256](#)
- ExposureTimeSelector\_Stage2, [256](#)
- ExposureTimeSelector\_Ultraviolet, [256](#)
- ExposureTimeSelector\_Yellow, [256](#)
- FileOpenMode\_Read, [256](#)
- FileOpenMode\_ReadWrite, [256](#)
- FileOpenMode\_Write, [256](#)
- FileOperationSelector\_Close, [257](#)
- FileOperationSelector\_Delete, [257](#)
- FileOperationSelector\_Open, [257](#)
- FileOperationSelector\_Read, [257](#)
- FileOperationSelector\_Write, [257](#)
- FileOperationStatus\_Failure, [257](#)
- FileOperationStatus\_Overflow, [257](#)
- FileOperationStatus\_Success, [257](#)
- FileSelector\_SerialPort0, [257](#)
- FileSelector\_UserFile1, [257](#)
- FileSelector\_UserSet0, [257](#)
- FileSelector\_UserSet1, [257](#)
- FileSelector\_UserSetDefault, [257](#)
- GainAuto\_Continuous, [259](#)
- GainAuto\_Off, [259](#)
- GainAuto\_Once, [259](#)
- GainAutoBalance\_Continuous, [259](#)

- GainAutoBalance\_Off, [259](#)
- GainAutoBalance\_Once, [259](#)
- GainSelector\_All, [259](#)
- GevCCP\_ControlAccess, [260](#)
- GevCCP\_ExclusiveAccess, [260](#)
- GevCCP\_OpenAccess, [260](#)
- GevCurrentPhysicalLinkConfiguration\_DynamicLAG, [260](#)
- GevCurrentPhysicalLinkConfiguration\_MultiLink, [260](#)
- GevCurrentPhysicalLinkConfiguration\_SingleLink, [260](#)
- GevCurrentPhysicalLinkConfiguration\_StaticLAG, [260](#)
- GevGVCPExtendedStatusCodesSelector\_Version1\_1, [260](#)
- GevGVCPExtendedStatusCodesSelector\_Version2\_0, [260](#)
- GevGVSPExtendedIDMode\_Off, [261](#)
- GevGVSPExtendedIDMode\_On, [261](#)
- GevIEEE1588ClockAccuracy\_Unknown, [261](#)
- GevIEEE1588Mode\_Auto, [261](#)
- GevIEEE1588Mode\_SlaveOnly, [261](#)
- GevIEEE1588Status\_Disabled, [261](#)
- GevIEEE1588Status\_Faulty, [261](#)
- GevIEEE1588Status\_Initializing, [261](#)
- GevIEEE1588Status\_Listening, [262](#)
- GevIEEE1588Status\_Master, [262](#)
- GevIEEE1588Status\_Passive, [262](#)
- GevIEEE1588Status\_PreMaster, [262](#)
- GevIEEE1588Status\_Slave, [262](#)
- GevIEEE1588Status\_Uncalibrated, [262](#)
- GevIPConfigurationStatus\_DHCP, [262](#)
- GevIPConfigurationStatus\_ForceIP, [262](#)
- GevIPConfigurationStatus\_LLA, [262](#)
- GevIPConfigurationStatus\_None, [262](#)
- GevIPConfigurationStatus\_PersistentIP, [262](#)
- GevPhysicalLinkConfiguration\_DynamicLAG, [262](#)
- GevPhysicalLinkConfiguration\_MultiLink, [262](#)
- GevPhysicalLinkConfiguration\_SingleLink, [262](#)
- GevPhysicalLinkConfiguration\_StaticLAG, [262](#)
- GevSupportedOptionSelector\_Action, [263](#)
- GevSupportedOptionSelector\_CCPApplicationSocket, [263](#)
- GevSupportedOptionSelector\_CommandsConcatenation, [263](#)
- GevSupportedOptionSelector\_DiscoveryAckDelay, [263](#)
- GevSupportedOptionSelector\_DiscoveryAckDelayWritable, [263](#)
- GevSupportedOptionSelector\_Event, [263](#)
- GevSupportedOptionSelector\_EventData, [263](#)
- GevSupportedOptionSelector\_ExtendedStatusCodes, [263](#)
- GevSupportedOptionSelector\_HeartbeatDisable, [263](#)
- GevSupportedOptionSelector\_IPConfigurationDHCP, [263](#)
- GevSupportedOptionSelector\_IPConfigurationLLA, [263](#)
- GevSupportedOptionSelector\_IPConfigurationPersistentIP, [263](#)
- GevSupportedOptionSelector\_LinkSpeed, [263](#)
- GevSupportedOptionSelector\_ManifestTable, [263](#)
- GevSupportedOptionSelector\_MessageChannelSourceSocket, [263](#)
- GevSupportedOptionSelector\_PacketResend, [263](#)
- GevSupportedOptionSelector\_PendingAck, [263](#)
- GevSupportedOptionSelector\_SerialNumber, [263](#)
- GevSupportedOptionSelector\_StreamChannelSourceSocket, [263](#)
- GevSupportedOptionSelector\_TestData, [263](#)
- GevSupportedOptionSelector\_UserDefinedName, [263](#)
- GevSupportedOptionSelector\_WriteMem, [263](#)
- ImageComponentSelector\_Color, [263](#)
- ImageComponentSelector\_Confidence, [264](#)
- ImageComponentSelector\_Disparity, [264](#)
- ImageComponentSelector\_Infrared, [263](#)
- ImageComponentSelector\_Intensity, [263](#)
- ImageComponentSelector\_Range, [263](#)
- ImageComponentSelector\_Scatter, [264](#)
- ImageComponentSelector\_Ultraviolet, [263](#)
- ImageCompressionJPEGFormatOption\_BaselineOptimized, [264](#)
- ImageCompressionJPEGFormatOption\_BaselineStandard, [264](#)
- ImageCompressionJPEGFormatOption\_Lossless, [264](#)
- ImageCompressionJPEGFormatOption\_Progressive, [264](#)
- ImageCompressionMode\_Lossless, [265](#)
- ImageCompressionMode\_Off, [265](#)
- ImageCompressionRateOption\_FixBitrate, [265](#)
- ImageCompressionRateOption\_FixQuality, [265](#)
- LineFormat\_LVDS, [265](#)
- LineFormat\_NoConnect, [265](#)
- LineFormat\_OpenDrain, [265](#)
- LineFormat\_OptoCoupled, [265](#)
- LineFormat\_RS422, [265](#)
- LineFormat\_TriState, [265](#)
- LineFormat\_TTL, [265](#)
- LineInputFilterSelector\_Debounce, [266](#)
- LineInputFilterSelector\_Deglitch, [266](#)
- LineMode\_Input, [266](#)
- LineMode\_Output, [266](#)
- LineSelector\_Line0, [266](#)
- LineSelector\_Line1, [266](#)
- LineSelector\_Line2, [266](#)
- LineSelector\_Line3, [266](#)
- LineSource\_AllPixel, [267](#)
- LineSource\_AnyPixel, [267](#)
- LineSource\_Counter0Active, [267](#)
- LineSource\_Counter1Active, [267](#)
- LineSource\_ExposureActive, [267](#)
- LineSource\_FrameTriggerWait, [267](#)



- LineSource\_Line0, [267](#)
- LineSource\_Line1, [267](#)
- LineSource\_Line2, [267](#)
- LineSource\_Line3, [267](#)
- LineSource\_LogicBlock0, [267](#)
- LineSource\_LogicBlock1, [267](#)
- LineSource\_Off, [267](#)
- LineSource\_PPSSignal, [267](#)
- LineSource\_SerialPort0, [267](#)
- LineSource\_UserOutput0, [267](#)
- LineSource\_UserOutput1, [267](#)
- LineSource\_UserOutput2, [267](#)
- LineSource\_UserOutput3, [267](#)
- LogicBlockLUTInputActivation\_AnyEdge, [267](#)
- LogicBlockLUTInputActivation\_FallingEdge, [267](#)
- LogicBlockLUTInputActivation\_LevelHigh, [267](#)
- LogicBlockLUTInputActivation\_LevelLow, [267](#)
- LogicBlockLUTInputActivation\_RisingEdge, [267](#)
- LogicBlockLUTInputSelector\_Input0, [268](#)
- LogicBlockLUTInputSelector\_Input1, [268](#)
- LogicBlockLUTInputSelector\_Input2, [268](#)
- LogicBlockLUTInputSelector\_Input3, [268](#)
- LogicBlockLUTInputSource\_AcquisitionActive, [268](#)
- LogicBlockLUTInputSource\_Counter0End, [268](#)
- LogicBlockLUTInputSource\_Counter0Start, [268](#)
- LogicBlockLUTInputSource\_Counter1End, [268](#)
- LogicBlockLUTInputSource\_Counter1Start, [268](#)
- LogicBlockLUTInputSource\_ExposureEnd, [268](#)
- LogicBlockLUTInputSource\_ExposureStart, [268](#)
- LogicBlockLUTInputSource\_FrameTriggerWait, [268](#)
- LogicBlockLUTInputSource\_Line0, [268](#)
- LogicBlockLUTInputSource\_Line1, [268](#)
- LogicBlockLUTInputSource\_Line2, [268](#)
- LogicBlockLUTInputSource\_Line3, [268](#)
- LogicBlockLUTInputSource\_LogicBlock0, [268](#)
- LogicBlockLUTInputSource\_LogicBlock1, [268](#)
- LogicBlockLUTInputSource\_UserOutput0, [268](#)
- LogicBlockLUTInputSource\_UserOutput1, [268](#)
- LogicBlockLUTInputSource\_UserOutput2, [268](#)
- LogicBlockLUTInputSource\_UserOutput3, [268](#)
- LogicBlockLUTInputSource\_Zero, [268](#)
- LogicBlockLUTSelector\_Enable, [269](#)
- LogicBlockLUTSelector\_Value, [269](#)
- LogicBlockSelector\_LogicBlock0, [269](#)
- LogicBlockSelector\_LogicBlock1, [269](#)
- LUTSelector\_LUT1, [269](#)
- NUM\_ACQUISITIONMODE, [220](#)
- NUM\_ACQUISITIONSTATUSSELECTOR, [220](#)
- NUM\_ACTIONUNCONDITIONALMODE, [220](#)
- NUM\_ADCBITDEPTH, [221](#)
- NUM\_AUTOALGORITHMSELECTOR, [221](#)
- NUM\_AUTOEXPOSURECONTROLPRIORITY, [221](#)
- NUM\_AUTOEXPOSURELIGHTINGMODE, [222](#)
- NUM\_AUTOEXPOSUREMETERINGMODE, [222](#)
- NUM\_AUTOEXPOSURETARGETGREYVALUEAUTO, [222](#)
- NUM\_BALANCERATIOSELECTOR, [223](#)
- NUM\_BALANCEWHITEAUTO, [223](#)
- NUM\_BALANCEWHITEAUTOPROFILE, [223](#)
- NUM\_BINNINGHORIZONTALMODE, [224](#)
- NUM\_BINNINGSELECTOR, [224](#)
- NUM\_BINNINGVERTICALMODE, [224](#)
- NUM\_BLACKLEVELAUTO, [225](#)
- NUM\_BLACKLEVELAUTOBALANCE, [225](#)
- NUM\_BLACKLEVELSELECTOR, [225](#)
- NUM\_CHUNKBLACKLEVELSELECTOR, [226](#)
- NUM\_CHUNKCOUNTERSELECTOR, [226](#)
- NUM\_CHUNKENCODERSELECTOR, [226](#)
- NUM\_CHUNKENCODERSTATUS, [227](#)
- NUM\_CHUNKEXPOSURETIMESELECTOR, [227](#)
- NUM\_CHUNKGAINSELECTOR, [227](#)
- NUM\_CHUNKIMAGECOMPONENT, [228](#)
- NUM\_CHUNKPIXELFORMAT, [228](#)
- NUM\_CHUNKREGIONID, [229](#)
- NUM\_CHUNKSCAN3DCOORDINATEREFERENCESELECTOR, [229](#)
- NUM\_CHUNKSCAN3DCOORDINATESELECTOR, [229](#)
- NUM\_CHUNKSCAN3DCOORDINATESYSTEM, [230](#)
- NUM\_CHUNKSCAN3DCOORDINATESYSTEMREFERENCE, [230](#)
- NUM\_CHUNKSCAN3DCOORDINATETRANSFORMSELECTOR, [230](#)
- NUM\_CHUNKSCAN3DDISTANCEUNIT, [231](#)
- NUM\_CHUNKSCAN3DOUTPUTMODE, [232](#)
- NUM\_CHUNKSELECTOR, [232](#)
- NUM\_CHUNKSOURCEID, [233](#)
- NUM\_CHUNKTIMERSELECTOR, [233](#)
- NUM\_CHUNKTRANSFERSTREAMID, [233](#)
- NUM\_CLCONFIGURATION, [234](#)
- NUM\_CLTIMESLOTSCOUNT, [234](#)
- NUM\_COLORTRANSFORMATIONSELECTOR, [234](#)
- NUM\_COLORTRANSFORMATIONVALUESELECTOR, [235](#)
- NUM\_COUNTEREVENTACTIVATION, [235](#)
- NUM\_COUNTEREVENTSOURCE, [236](#)
- NUM\_COUNTERRESETACTIVATION, [236](#)
- NUM\_COUNTERRESETSOURCE, [237](#)
- NUM\_COUNTERSELECTOR, [237](#)
- NUM\_COUNTERSTATUS, [237](#)
- NUM\_COUNTERTRIGGERACTIVATION, [238](#)
- NUM\_COUNTERTRIGGERSOURCE, [238](#)
- NUM\_CXPCONNECTIONTESTMODE, [239](#)
- NUM\_CXPLINKCONFIGURATION, [240](#)
- NUM\_CXPLINKCONFIGURATIONPREFERRED, [241](#)
- NUM\_CXPLINKCONFIGURATIONSTATUS, [242](#)
- NUM\_CXPPOCXPSTATUS, [242](#)
- NUM\_DECIMATIONHORIZONTALMODE, [242](#)
- NUM\_DECIMATIONSELECTOR, [243](#)
- NUM\_DECIMATIONVERTICALMODE, [243](#)
- NUM\_DEFECTCORRECTIONMODE, [243](#)

- NUM\_DEINTERLACING, [244](#)
- NUM\_DEVICECHARACTERSET, [244](#)
- NUM\_DEVICECLOCKSELECTOR, [244](#)
- NUM\_DEVICECONNECTIONSTATUS, [244](#)
- NUM\_DEVICEINDICATORMODE, [245](#)
- NUM\_DEVICELINKHEARTBEATMODE, [245](#)
- NUM\_DEVICELINKTHROUGHPUTLIMITMODE, [245](#)
- NUM\_DEVICEPOWERSUPPLYSELECTOR, [246](#)
- NUM\_DEVICEREGISTERSENDIANNES, [246](#)
- NUM\_DEVICESCANTYPE, [246](#)
- NUM\_DEVICESERIALPORTBAUDRATE, [247](#)
- NUM\_DEVICESERIALPORTSELECTOR, [247](#)
- NUM\_DEVICESTREAMCHANNELENDIANNES, [247](#)
- NUM\_DEVICESTREAMCHANNELTYPE, [248](#)
- NUM\_DEVICETAPGEOMETRY, [249](#)
- NUM\_DEVICETEMPERATURESELECTOR, [249](#)
- NUM\_DEVICETLTYPE, [249](#)
- NUM\_DEVICETYPE, [250](#)
- NUM\_ENCODERMODE, [250](#)
- NUM\_ENCODEROUTPUTMODE, [251](#)
- NUM\_ENCODERRESETACTIVATION, [251](#)
- NUM\_ENCODERRESETSOURCE, [252](#)
- NUM\_ENCODERSELECTOR, [252](#)
- NUM\_ENCODERSOURCEA, [253](#)
- NUM\_ENCODERSOURCEB, [253](#)
- NUM\_ENCODERSTATUS, [253](#)
- NUM\_EVENTNOTIFICATION, [254](#)
- NUM\_EVENTSELECTOR, [254](#)
- NUM\_EXPOSUREACTIVEMODE, [254](#)
- NUM\_EXPOSUREAUTO, [255](#)
- NUM\_EXPOSUREMODE, [255](#)
- NUM\_EXPOSURETIMEMODE, [255](#)
- NUM\_EXPOSURETIMESELECTOR, [256](#)
- NUM\_FILEOPENMODE, [256](#)
- NUM\_FILEOPERATIONSELECTOR, [257](#)
- NUM\_FILEOPERATIONSTATUS, [257](#)
- NUM\_FILESELECTOR, [257](#)
- NUM\_GAINAUTO, [259](#)
- NUM\_GAINAUTOBALANCE, [259](#)
- NUM\_GAINSELECTOR, [259](#)
- NUM\_GEVCPP, [260](#)
- NUM\_GEVCURRENTPHYSICALLINKCONFIGURATION, [260](#)
- NUM\_GEVGVCPEXTENDEDSTATUSCODESSELECTOR, [260](#)
- NUM\_GEVGVSPEXTENDEDIDMODE, [261](#)
- NUM\_GEVEEEE1588CLOCKACCURACY, [261](#)
- NUM\_GEVEEEE1588MODE, [261](#)
- NUM\_GEVEEEE1588STATUS, [262](#)
- NUM\_GEVIPCONFIGURATIONSTATUS, [262](#)
- NUM\_GEVPHYSCALLINKCONFIGURATION, [262](#)
- NUM\_GEVSUPPORTEDOPTIONSELECTOR, [263](#)
- NUM\_IMAGECOMPONENTSELECTOR, [264](#)
- NUM\_IMAGECOMPRESSIONJPEGFORMATOPTION, [264](#)
- NUM\_IMAGECOMPRESSIONMODE, [265](#)
- NUM\_IMAGECOMPRESSIONRATEOPTION, [265](#)
- NUM\_LINEFORMAT, [265](#)
- NUM\_LINEINPUTFILTERSELECTOR, [266](#)
- NUM\_LINEMODE, [266](#)
- NUM\_LINESELECTOR, [266](#)
- NUM\_LINESOURCE, [267](#)
- NUM\_LOGICBLOCKLUTINPUTACTIVATION, [267](#)
- NUM\_LOGICBLOCKLUTINPUTSELECTOR, [268](#)
- NUM\_LOGICBLOCKLUTINPUTSOURCE, [268](#)
- NUM\_LOGICBLOCKLUTSELECTOR, [269](#)
- NUM\_LOGICBLOCKSELECTOR, [269](#)
- NUM\_LUTSELECTOR, [269](#)
- NUM\_PIXELCOLORFILTER, [270](#)
- NUM\_PIXELFORMAT, [275](#)
- NUM\_PIXELFORMATINFOSELECTOR, [281](#)
- NUM\_PIXELSIZE, [282](#)
- NUM\_REGIONDESTINATION, [282](#)
- NUM\_REGIONMODE, [282](#)
- NUM\_REGIONSELECTOR, [283](#)
- NUM\_RGBTRANSFORMLIGHTSOURCE, [283](#)
- NUM\_SCAN3DCOORDINATEREFERENCESELECTOR, [284](#)
- NUM\_SCAN3DCOORDINATESELECTOR, [284](#)
- NUM\_SCAN3DCOORDINATESYSTEM, [284](#)
- NUM\_SCAN3DCOORDINATESYSTEMREFERENCE, [285](#)
- NUM\_SCAN3DCOORDINATETRANSFORMSELECTOR, [285](#)
- NUM\_SCAN3DDISTANCEUNIT, [285](#)
- NUM\_SCAN3DOUTPUTMODE, [286](#)
- NUM\_SENSORDIGITIZATIONTAPS, [287](#)
- NUM\_SENSORSHUTTERMODE, [287](#)
- NUM\_SENSORTAPS, [287](#)
- NUM\_SEQUENCERCONFIGURATIONMODE, [288](#)
- NUM\_SEQUENCERCONFIGURATIONVALID, [288](#)
- NUM\_SEQUENCERMODE, [288](#)
- NUM\_SEQUENCERSETVALID, [289](#)
- NUM\_SEQUENCERTRIGGERACTIVATION, [289](#)
- NUM\_SEQUENCERTRIGGERSOURCE, [289](#)
- NUM\_SERIALPORTBAUDRATE, [290](#)
- NUM\_SERIALPORTPARITY, [290](#)
- NUM\_SERIALPORTSELECTOR, [290](#)
- NUM\_SERIALPORTSOURCE, [291](#)
- NUM\_SERIALPORTSTOPBITS, [291](#)
- NUM\_SOFTWARESIGNALSELECTOR, [291](#)
- NUM\_SOURCESELECTOR, [292](#)
- NUM\_TESTPATTERN, [292](#)
- NUM\_TESTPATTERNGENERATORSELECTOR, [292](#)
- NUM\_TIMERSELECTOR, [293](#)
- NUM\_TIMERSTATUS, [293](#)
- NUM\_TIMERTRIGGERACTIVATION, [293](#)
- NUM\_TIMERTRIGGERSOURCE, [295](#)
- NUM\_TRANSFERCOMPONENTSELECTOR, [295](#)
- NUM\_TRANSFERCONTROLMODE, [296](#)
- NUM\_TRANSFEROPERATIONMODE, [296](#)

NUM\_TRANSFERQUEUEMODE, [296](#)  
NUM\_TRANSFERSELECTOR, [296](#)  
NUM\_TRANSFERSTATUSSELECTOR, [297](#)  
NUM\_TRANSFERTRIGGERACTIVATION, [297](#)  
NUM\_TRANSFERTRIGGERMODE, [298](#)  
NUM\_TRANSFERTRIGGERSELECTOR, [298](#)  
NUM\_TRANSFERTRIGGERSOURCE, [299](#)  
NUM\_TRIGGERACTIVATION, [299](#)  
NUM\_TRIGGERMODE, [300](#)  
NUM\_TRIGGEROVERLAP, [300](#)  
NUM\_TRIGGERSELECTOR, [300](#)  
NUM\_TRIGGERSOURCE, [301](#)  
NUM\_USEROUTPUTSELECTOR, [301](#)  
NUM\_USERSETDEFAULT, [302](#)  
NUM\_USERSETSELECTOR, [302](#)  
NUM\_WHITECLIPSELECTOR, [302](#)  
PixelFormat\_BayerBG, [270](#)  
PixelFormat\_BayerGB, [270](#)  
PixelFormat\_BayerGR, [270](#)  
PixelFormat\_BayerRG, [269](#)  
PixelFormat\_None, [269](#)  
PixelFormat\_B10, [272](#)  
PixelFormat\_B12, [272](#)  
PixelFormat\_B12\_Jpeg, [275](#)  
PixelFormat\_B16, [272](#)  
PixelFormat\_B8, [272](#)  
PixelFormat\_BayerBG10, [271](#)  
PixelFormat\_BayerBG10p, [271](#)  
PixelFormat\_BayerBG10Packed, [271](#)  
PixelFormat\_BayerBG12, [271](#)  
PixelFormat\_BayerBG12p, [270](#)  
PixelFormat\_BayerBG12Packed, [270](#)  
PixelFormat\_BayerBG16, [270](#)  
PixelFormat\_BayerBG8, [270](#)  
PixelFormat\_BayerGB10, [271](#)  
PixelFormat\_BayerGB10p, [271](#)  
PixelFormat\_BayerGB10Packed, [271](#)  
PixelFormat\_BayerGB12, [271](#)  
PixelFormat\_BayerGB12p, [270](#)  
PixelFormat\_BayerGB12Packed, [270](#)  
PixelFormat\_BayerGB16, [270](#)  
PixelFormat\_BayerGB8, [270](#)  
PixelFormat\_BayerGR10, [271](#)  
PixelFormat\_BayerGR10p, [271](#)  
PixelFormat\_BayerGR10Packed, [270](#)  
PixelFormat\_BayerGR12, [271](#)  
PixelFormat\_BayerGR12p, [270](#)  
PixelFormat\_BayerGR12Packed, [270](#)  
PixelFormat\_BayerGR16, [270](#)  
PixelFormat\_BayerGR8, [270](#)  
PixelFormat\_BayerRG10, [271](#)  
PixelFormat\_BayerRG10p, [271](#)  
PixelFormat\_BayerRG10Packed, [270](#)  
PixelFormat\_BayerRG12, [271](#)  
PixelFormat\_BayerRG12p, [270](#)  
PixelFormat\_BayerRG12Packed, [270](#)  
PixelFormat\_BayerRG16, [270](#)  
PixelFormat\_BayerRG8, [270](#)  
PixelFormat\_BayerRG10p, [275](#)  
PixelFormat\_BayerRG12p, [275](#)  
PixelFormat\_BayerRG16, [275](#)  
PixelFormat\_BayerRG8, [270](#)  
PixelFormat\_BiColorBGRG10, [273](#)  
PixelFormat\_BiColorBGRG10p, [273](#)  
PixelFormat\_BiColorBGRG12, [273](#)  
PixelFormat\_BiColorBGRG12p, [273](#)  
PixelFormat\_BiColorBGRG8, [273](#)  
PixelFormat\_BiColorRGBG10, [273](#)  
PixelFormat\_BiColorRGBG10p, [273](#)  
PixelFormat\_BiColorRGBG12, [273](#)  
PixelFormat\_BiColorRGBG12p, [273](#)  
PixelFormat\_BiColorRGBG8, [273](#)  
PixelFormat\_Confidence1, [273](#)  
PixelFormat\_Confidence16, [273](#)  
PixelFormat\_Confidence1p, [273](#)  
PixelFormat\_Confidence32f, [273](#)  
PixelFormat\_Confidence8, [273](#)  
PixelFormat\_Coord3D\_A10p, [273](#)  
PixelFormat\_Coord3D\_A12p, [273](#)  
PixelFormat\_Coord3D\_A16, [273](#)  
PixelFormat\_Coord3D\_A32f, [273](#)  
PixelFormat\_Coord3D\_A8, [273](#)  
PixelFormat\_Coord3D\_ABC10p, [272](#)  
PixelFormat\_Coord3D\_ABC10p\_Planar, [272](#)  
PixelFormat\_Coord3D\_ABC12p, [272](#)  
PixelFormat\_Coord3D\_ABC12p\_Planar, [272](#)  
PixelFormat\_Coord3D\_ABC16, [272](#)  
PixelFormat\_Coord3D\_ABC16\_Planar, [272](#)  
PixelFormat\_Coord3D\_ABC32f, [272](#)  
PixelFormat\_Coord3D\_ABC32f\_Planar, [272](#)  
PixelFormat\_Coord3D\_ABC8, [272](#)  
PixelFormat\_Coord3D\_ABC8\_Planar, [272](#)  
PixelFormat\_Coord3D\_AC10p, [272](#)  
PixelFormat\_Coord3D\_AC10p\_Planar, [272](#)  
PixelFormat\_Coord3D\_AC12p, [272](#)  
PixelFormat\_Coord3D\_AC12p\_Planar, [272](#)  
PixelFormat\_Coord3D\_AC16, [272](#)  
PixelFormat\_Coord3D\_AC16\_Planar, [272](#)  
PixelFormat\_Coord3D\_AC32f, [272](#)  
PixelFormat\_Coord3D\_AC32f\_Planar, [272](#)  
PixelFormat\_Coord3D\_AC8, [272](#)

PixelFormat\_Coord3D\_AC8\_Planar, [272](#)  
 PixelFormat\_Coord3D\_B10p, [273](#)  
 PixelFormat\_Coord3D\_B12p, [273](#)  
 PixelFormat\_Coord3D\_B16, [273](#)  
 PixelFormat\_Coord3D\_B32f, [273](#)  
 PixelFormat\_Coord3D\_B8, [273](#)  
 PixelFormat\_Coord3D\_C10p, [273](#)  
 PixelFormat\_Coord3D\_C12p, [273](#)  
 PixelFormat\_Coord3D\_C16, [273](#)  
 PixelFormat\_Coord3D\_C32f, [273](#)  
 PixelFormat\_Coord3D\_C8, [273](#)  
 PixelFormat\_G10, [272](#)  
 PixelFormat\_G12, [272](#)  
 PixelFormat\_G16, [272](#)  
 PixelFormat\_G8, [272](#)  
 PixelFormat\_GB12\_Jpeg, [275](#)  
 PixelFormat\_GR12\_Jpeg, [275](#)  
 PixelFormat\_JPEGColor8, [275](#)  
 PixelFormat\_JPEGMono8, [275](#)  
 PixelFormat\_LLCBayerRG8, [275](#)  
 PixelFormat\_LLCMono8, [275](#)  
 PixelFormat\_Mono10, [271](#)  
 PixelFormat\_Mono10p, [271](#)  
 PixelFormat\_Mono10Packed, [270](#)  
 PixelFormat\_Mono12, [271](#)  
 PixelFormat\_Mono12p, [270](#)  
 PixelFormat\_Mono12Packed, [270](#)  
 PixelFormat\_Mono14, [271](#)  
 PixelFormat\_Mono16, [270](#)  
 PixelFormat\_Mono16s, [271](#)  
 PixelFormat\_Mono1p, [271](#)  
 PixelFormat\_Mono2p, [271](#)  
 PixelFormat\_Mono32f, [271](#)  
 PixelFormat\_Mono4p, [271](#)  
 PixelFormat\_Mono8, [270](#)  
 PixelFormat\_Mono8s, [271](#)  
 PixelFormat\_Polarized10p, [275](#)  
 PixelFormat\_Polarized12p, [275](#)  
 PixelFormat\_Polarized16, [275](#)  
 PixelFormat\_Polarized8, [275](#)  
 PixelFormat\_R10, [272](#)  
 PixelFormat\_R12, [272](#)  
 PixelFormat\_R12\_Jpeg, [275](#)  
 PixelFormat\_R16, [272](#)  
 PixelFormat\_R8, [272](#)  
 PixelFormat\_Raw16, [275](#)  
 PixelFormat\_Raw8, [275](#)  
 PixelFormat\_RGB10, [271](#)  
 PixelFormat\_RGB10\_Planar, [271](#)  
 PixelFormat\_RGB10p, [271](#)  
 PixelFormat\_RGB10p32, [271](#)  
 PixelFormat\_RGB12, [271](#)  
 PixelFormat\_RGB12\_Planar, [271](#)  
 PixelFormat\_RGB12p, [271](#)  
 PixelFormat\_RGB14, [271](#)  
 PixelFormat\_RGB16, [271](#)  
 PixelFormat\_RGB16\_Planar, [271](#)  
 PixelFormat\_RGB16s, [271](#)  
 PixelFormat\_RGB32f, [271](#)  
 PixelFormat\_RGB565p, [272](#)  
 PixelFormat\_RGB8, [271](#)  
 PixelFormat\_RGB8\_Planar, [271](#)  
 PixelFormat\_RGB8Packed, [270](#)  
 PixelFormat\_RGBa10, [271](#)  
 PixelFormat\_RGBa10p, [271](#)  
 PixelFormat\_RGBa12, [271](#)  
 PixelFormat\_RGBa12p, [271](#)  
 PixelFormat\_RGBa14, [271](#)  
 PixelFormat\_RGBa16, [271](#)  
 PixelFormat\_RGBa32f, [272](#)  
 PixelFormat\_RGBa8, [271](#)  
 PixelFormat\_SCF1WBWG10, [273](#)  
 PixelFormat\_SCF1WBWG10p, [273](#)  
 PixelFormat\_SCF1WBWG12, [273](#)  
 PixelFormat\_SCF1WBWG12p, [273](#)  
 PixelFormat\_SCF1WBWG14, [273](#)  
 PixelFormat\_SCF1WBWG16, [273](#)  
 PixelFormat\_SCF1WBWG8, [273](#)  
 PixelFormat\_SCF1WGWB10, [273](#)  
 PixelFormat\_SCF1WGWB10p, [273](#)  
 PixelFormat\_SCF1WGWB12, [273](#)  
 PixelFormat\_SCF1WGWB12p, [273](#)  
 PixelFormat\_SCF1WGWB14, [273](#)  
 PixelFormat\_SCF1WGWB16, [273](#)  
 PixelFormat\_SCF1WGWB8, [273](#)  
 PixelFormat\_SCF1WGWR10, [273](#)  
 PixelFormat\_SCF1WGWR10p, [273](#)  
 PixelFormat\_SCF1WGWR12, [274](#)  
 PixelFormat\_SCF1WGWR12p, [274](#)  
 PixelFormat\_SCF1WGWR14, [274](#)  
 PixelFormat\_SCF1WGWR16, [274](#)  
 PixelFormat\_SCF1WGWR8, [273](#)  
 PixelFormat\_SCF1WRWG10, [274](#)  
 PixelFormat\_SCF1WRWG10p, [274](#)  
 PixelFormat\_SCF1WRWG12, [274](#)  
 PixelFormat\_SCF1WRWG12p, [274](#)  
 PixelFormat\_SCF1WRWG14, [274](#)  
 PixelFormat\_SCF1WRWG16, [274](#)  
 PixelFormat\_SCF1WRWG8, [274](#)  
 PixelFormat\_YCbCr10\_CbYCr, [274](#)  
 PixelFormat\_YCbCr10p\_CbYCr, [274](#)  
 PixelFormat\_YCbCr12\_CbYCr, [274](#)  
 PixelFormat\_YCbCr12p\_CbYCr, [274](#)  
 PixelFormat\_YCbCr411\_8, [270](#)  
 PixelFormat\_YCbCr411\_8\_CbYYCrYY, [274](#)  
 PixelFormat\_YCbCr422\_10, [274](#)  
 PixelFormat\_YCbCr422\_10\_CbYCrY, [274](#)  
 PixelFormat\_YCbCr422\_10p, [274](#)  
 PixelFormat\_YCbCr422\_10p\_CbYCrY, [274](#)  
 PixelFormat\_YCbCr422\_12, [274](#)  
 PixelFormat\_YCbCr422\_12\_CbYCrY, [274](#)  
 PixelFormat\_YCbCr422\_12p, [274](#)  
 PixelFormat\_YCbCr422\_12p\_CbYCrY, [274](#)  
 PixelFormat\_YCbCr422\_8, [270](#)  
 PixelFormat\_YCbCr422\_8\_CbYCrY, [274](#)  
 PixelFormat\_YCbCr601\_10\_CbYCr, [274](#)



- PixelFormat\_YCbCr601\_10p\_CbYCr, [274](#)
- PixelFormat\_YCbCr601\_12\_CbYCr, [274](#)
- PixelFormat\_YCbCr601\_12p\_CbYCr, [274](#)
- PixelFormat\_YCbCr601\_411\_8\_CbYYCrYY, [274](#)
- PixelFormat\_YCbCr601\_422\_10, [274](#)
- PixelFormat\_YCbCr601\_422\_10\_CbYCrY, [274](#)
- PixelFormat\_YCbCr601\_422\_10p, [274](#)
- PixelFormat\_YCbCr601\_422\_10p\_CbYCrY, [274](#)
- PixelFormat\_YCbCr601\_422\_12, [274](#)
- PixelFormat\_YCbCr601\_422\_12\_CbYCrY, [274](#)
- PixelFormat\_YCbCr601\_422\_12p, [274](#)
- PixelFormat\_YCbCr601\_422\_12p\_CbYCrY, [274](#)
- PixelFormat\_YCbCr601\_422\_8, [274](#)
- PixelFormat\_YCbCr601\_422\_8\_CbYCrY, [274](#)
- PixelFormat\_YCbCr601\_8\_CbYCr, [274](#)
- PixelFormat\_YCbCr709\_10\_CbYCr, [274](#)
- PixelFormat\_YCbCr709\_10p\_CbYCr, [274](#)
- PixelFormat\_YCbCr709\_12\_CbYCr, [274](#)
- PixelFormat\_YCbCr709\_12p\_CbYCr, [274](#)
- PixelFormat\_YCbCr709\_411\_8\_CbYYCrYY, [275](#)
- PixelFormat\_YCbCr709\_422\_10, [275](#)
- PixelFormat\_YCbCr709\_422\_10\_CbYCrY, [275](#)
- PixelFormat\_YCbCr709\_422\_10p, [275](#)
- PixelFormat\_YCbCr709\_422\_10p\_CbYCrY, [275](#)
- PixelFormat\_YCbCr709\_422\_12, [275](#)
- PixelFormat\_YCbCr709\_422\_12\_CbYCrY, [275](#)
- PixelFormat\_YCbCr709\_422\_12p, [275](#)
- PixelFormat\_YCbCr709\_422\_12p\_CbYCrY, [275](#)
- PixelFormat\_YCbCr709\_422\_8, [275](#)
- PixelFormat\_YCbCr709\_422\_8\_CbYCrY, [275](#)
- PixelFormat\_YCbCr709\_8\_CbYCr, [274](#)
- PixelFormat\_YCbCr8, [270](#)
- PixelFormat\_YCbCr8\_CbYCr, [274](#)
- PixelFormat\_YUV411\_8\_UYVYY, [275](#)
- PixelFormat\_YUV411Packed, [270](#)
- PixelFormat\_YUV422\_8, [275](#)
- PixelFormat\_YUV422\_8\_UYVY, [275](#)
- PixelFormat\_YUV422Packed, [270](#)
- PixelFormat\_YUV444Packed, [270](#)
- PixelFormat\_YUV8\_UYV, [275](#)
- PixelFormatInfoSelector\_B10, [277](#)
- PixelFormatInfoSelector\_B12, [277](#)
- PixelFormatInfoSelector\_B16, [277](#)
- PixelFormatInfoSelector\_B8, [277](#)
- PixelFormatInfoSelector\_BayerBG10, [276](#)
- PixelFormatInfoSelector\_BayerBG10p, [276](#)
- PixelFormatInfoSelector\_BayerBG12, [276](#)
- PixelFormatInfoSelector\_BayerBG12p, [276](#)
- PixelFormatInfoSelector\_BayerBG16, [276](#)
- PixelFormatInfoSelector\_BayerBG8, [276](#)
- PixelFormatInfoSelector\_BayerGB10, [276](#)
- PixelFormatInfoSelector\_BayerGB10p, [276](#)
- PixelFormatInfoSelector\_BayerGB12, [276](#)
- PixelFormatInfoSelector\_BayerGB12p, [276](#)
- PixelFormatInfoSelector\_BayerGB16, [276](#)
- PixelFormatInfoSelector\_BayerGB8, [276](#)
- PixelFormatInfoSelector\_BayerGR10, [276](#)
- PixelFormatInfoSelector\_BayerGR10p, [276](#)
- PixelFormatInfoSelector\_BayerGR12, [276](#)
- PixelFormatInfoSelector\_BayerGR12p, [276](#)
- PixelFormatInfoSelector\_BayerGR16, [276](#)
- PixelFormatInfoSelector\_BayerGR8, [276](#)
- PixelFormatInfoSelector\_BayerRG10, [276](#)
- PixelFormatInfoSelector\_BayerRG10p, [276](#)
- PixelFormatInfoSelector\_BayerRG12, [276](#)
- PixelFormatInfoSelector\_BayerRG12p, [276](#)
- PixelFormatInfoSelector\_BayerRG16, [276](#)
- PixelFormatInfoSelector\_BayerRG8, [276](#)
- PixelFormatInfoSelector\_BayerRGPolarized10p, [281](#)
- PixelFormatInfoSelector\_BayerRGPolarized12p, [281](#)
- PixelFormatInfoSelector\_BayerRGPolarized16, [281](#)
- PixelFormatInfoSelector\_BayerRGPolarized8, [281](#)
- PixelFormatInfoSelector\_BGR10, [277](#)
- PixelFormatInfoSelector\_BGR10p, [277](#)
- PixelFormatInfoSelector\_BGR12, [277](#)
- PixelFormatInfoSelector\_BGR12p, [277](#)
- PixelFormatInfoSelector\_BGR14, [277](#)
- PixelFormatInfoSelector\_BGR16, [277](#)
- PixelFormatInfoSelector\_BGR565p, [277](#)
- PixelFormatInfoSelector\_BGR8, [277](#)
- PixelFormatInfoSelector\_BGRa10, [277](#)
- PixelFormatInfoSelector\_BGRa10p, [277](#)
- PixelFormatInfoSelector\_BGRa12, [277](#)
- PixelFormatInfoSelector\_BGRa12p, [277](#)
- PixelFormatInfoSelector\_BGRa14, [277](#)
- PixelFormatInfoSelector\_BGRa16, [277](#)
- PixelFormatInfoSelector\_BGRa8, [277](#)
- PixelFormatInfoSelector\_BiColorBGRG10, [278](#)
- PixelFormatInfoSelector\_BiColorBGRG10p, [278](#)
- PixelFormatInfoSelector\_BiColorBGRG12, [278](#)
- PixelFormatInfoSelector\_BiColorBGRG12p, [278](#)
- PixelFormatInfoSelector\_BiColorBGRG8, [278](#)
- PixelFormatInfoSelector\_BiColorRGBG10, [278](#)
- PixelFormatInfoSelector\_BiColorRGBG10p, [278](#)
- PixelFormatInfoSelector\_BiColorRGBG12, [278](#)
- PixelFormatInfoSelector\_BiColorRGBG12p, [278](#)
- PixelFormatInfoSelector\_BiColorRGBG8, [278](#)
- PixelFormatInfoSelector\_Confidence1, [278](#)
- PixelFormatInfoSelector\_Confidence16, [278](#)
- PixelFormatInfoSelector\_Confidence1p, [278](#)
- PixelFormatInfoSelector\_Confidence32f, [278](#)
- PixelFormatInfoSelector\_Confidence8, [278](#)
- PixelFormatInfoSelector\_Coord3D\_A10p, [278](#)
- PixelFormatInfoSelector\_Coord3D\_A12p, [278](#)
- PixelFormatInfoSelector\_Coord3D\_A16, [278](#)
- PixelFormatInfoSelector\_Coord3D\_A32f, [278](#)
- PixelFormatInfoSelector\_Coord3D\_A8, [278](#)
- PixelFormatInfoSelector\_Coord3D\_ABC10p, [277](#)
- PixelFormatInfoSelector\_Coord3D\_ABC10p\_Planar, [277](#)
- PixelFormatInfoSelector\_Coord3D\_ABC12p, [277](#)
- PixelFormatInfoSelector\_Coord3D\_ABC12p\_Planar, [277](#)

- PixelFormatInfoSelector\_Coord3D\_ABC16, [277](#)
- PixelFormatInfoSelector\_Coord3D\_ABC16\_Planar, [278](#)
- PixelFormatInfoSelector\_Coord3D\_ABC32f, [278](#)
- PixelFormatInfoSelector\_Coord3D\_ABC32f\_Planar, [278](#)
- PixelFormatInfoSelector\_Coord3D\_ABC8, [277](#)
- PixelFormatInfoSelector\_Coord3D\_ABC8\_Planar, [277](#)
- PixelFormatInfoSelector\_Coord3D\_AC10p, [278](#)
- PixelFormatInfoSelector\_Coord3D\_AC10p\_Planar, [278](#)
- PixelFormatInfoSelector\_Coord3D\_AC12p, [278](#)
- PixelFormatInfoSelector\_Coord3D\_AC12p\_Planar, [278](#)
- PixelFormatInfoSelector\_Coord3D\_AC16, [278](#)
- PixelFormatInfoSelector\_Coord3D\_AC16\_Planar, [278](#)
- PixelFormatInfoSelector\_Coord3D\_AC32f, [278](#)
- PixelFormatInfoSelector\_Coord3D\_AC32f\_Planar, [278](#)
- PixelFormatInfoSelector\_Coord3D\_AC8, [278](#)
- PixelFormatInfoSelector\_Coord3D\_AC8\_Planar, [278](#)
- PixelFormatInfoSelector\_Coord3D\_B10p, [278](#)
- PixelFormatInfoSelector\_Coord3D\_B12p, [278](#)
- PixelFormatInfoSelector\_Coord3D\_B16, [278](#)
- PixelFormatInfoSelector\_Coord3D\_B32f, [278](#)
- PixelFormatInfoSelector\_Coord3D\_B8, [278](#)
- PixelFormatInfoSelector\_Coord3D\_C10p, [278](#)
- PixelFormatInfoSelector\_Coord3D\_C12p, [278](#)
- PixelFormatInfoSelector\_Coord3D\_C16, [278](#)
- PixelFormatInfoSelector\_Coord3D\_C32f, [278](#)
- PixelFormatInfoSelector\_Coord3D\_C8, [278](#)
- PixelFormatInfoSelector\_G10, [277](#)
- PixelFormatInfoSelector\_G12, [277](#)
- PixelFormatInfoSelector\_G16, [277](#)
- PixelFormatInfoSelector\_G8, [277](#)
- PixelFormatInfoSelector\_JPEGColor8, [281](#)
- PixelFormatInfoSelector\_JPEGMono8, [281](#)
- PixelFormatInfoSelector\_LLCBayerRG8, [281](#)
- PixelFormatInfoSelector\_LLCMono8, [281](#)
- PixelFormatInfoSelector\_Mono10, [276](#)
- PixelFormatInfoSelector\_Mono10p, [276](#)
- PixelFormatInfoSelector\_Mono12, [276](#)
- PixelFormatInfoSelector\_Mono12p, [276](#)
- PixelFormatInfoSelector\_Mono14, [276](#)
- PixelFormatInfoSelector\_Mono16, [276](#)
- PixelFormatInfoSelector\_Mono16s, [276](#)
- PixelFormatInfoSelector\_Mono1p, [275](#)
- PixelFormatInfoSelector\_Mono2p, [276](#)
- PixelFormatInfoSelector\_Mono32f, [276](#)
- PixelFormatInfoSelector\_Mono4p, [276](#)
- PixelFormatInfoSelector\_Mono8, [276](#)
- PixelFormatInfoSelector\_Mono8s, [276](#)
- PixelFormatInfoSelector\_Polarized10p, [281](#)
- PixelFormatInfoSelector\_Polarized12p, [281](#)
- PixelFormatInfoSelector\_Polarized16, [281](#)
- PixelFormatInfoSelector\_Polarized8, [281](#)
- PixelFormatInfoSelector\_R10, [277](#)
- PixelFormatInfoSelector\_R12, [277](#)
- PixelFormatInfoSelector\_R16, [277](#)
- PixelFormatInfoSelector\_R8, [277](#)
- PixelFormatInfoSelector\_RGB10, [276](#)
- PixelFormatInfoSelector\_RGB10\_Planar, [276](#)
- PixelFormatInfoSelector\_RGB10p, [277](#)
- PixelFormatInfoSelector\_RGB10p32, [277](#)
- PixelFormatInfoSelector\_RGB12, [277](#)
- PixelFormatInfoSelector\_RGB12\_Planar, [277](#)
- PixelFormatInfoSelector\_RGB12p, [277](#)
- PixelFormatInfoSelector\_RGB14, [277](#)
- PixelFormatInfoSelector\_RGB16, [277](#)
- PixelFormatInfoSelector\_RGB16\_Planar, [277](#)
- PixelFormatInfoSelector\_RGB16s, [277](#)
- PixelFormatInfoSelector\_RGB32f, [277](#)
- PixelFormatInfoSelector\_RGB565p, [277](#)
- PixelFormatInfoSelector\_RGB8, [276](#)
- PixelFormatInfoSelector\_RGB8\_Planar, [276](#)
- PixelFormatInfoSelector\_RGBa10, [276](#)
- PixelFormatInfoSelector\_RGBa10p, [276](#)
- PixelFormatInfoSelector\_RGBa12, [276](#)
- PixelFormatInfoSelector\_RGBa12p, [276](#)
- PixelFormatInfoSelector\_RGBa14, [276](#)
- PixelFormatInfoSelector\_RGBa16, [276](#)
- PixelFormatInfoSelector\_RGBa32f, [277](#)
- PixelFormatInfoSelector\_RGBa8, [276](#)
- PixelFormatInfoSelector\_SCF1WBWG10, [278](#)
- PixelFormatInfoSelector\_SCF1WBWG10p, [279](#)
- PixelFormatInfoSelector\_SCF1WBWG12, [279](#)
- PixelFormatInfoSelector\_SCF1WBWG12p, [279](#)
- PixelFormatInfoSelector\_SCF1WBWG14, [279](#)
- PixelFormatInfoSelector\_SCF1WBWG16, [279](#)
- PixelFormatInfoSelector\_SCF1WBWG8, [278](#)
- PixelFormatInfoSelector\_SCF1WGWB10, [279](#)
- PixelFormatInfoSelector\_SCF1WGWB10p, [279](#)
- PixelFormatInfoSelector\_SCF1WGWB12, [279](#)
- PixelFormatInfoSelector\_SCF1WGWB12p, [279](#)
- PixelFormatInfoSelector\_SCF1WGWB14, [279](#)
- PixelFormatInfoSelector\_SCF1WGWB16, [279](#)
- PixelFormatInfoSelector\_SCF1WGWB8, [279](#)
- PixelFormatInfoSelector\_SCF1WGWR10, [279](#)
- PixelFormatInfoSelector\_SCF1WGWR10p, [279](#)
- PixelFormatInfoSelector\_SCF1WGWR12, [279](#)
- PixelFormatInfoSelector\_SCF1WGWR12p, [279](#)
- PixelFormatInfoSelector\_SCF1WGWR14, [279](#)
- PixelFormatInfoSelector\_SCF1WGWR16, [279](#)
- PixelFormatInfoSelector\_SCF1WGWR8, [279](#)
- PixelFormatInfoSelector\_SCF1WRWG10, [279](#)
- PixelFormatInfoSelector\_SCF1WRWG10p, [279](#)
- PixelFormatInfoSelector\_SCF1WRWG12, [279](#)
- PixelFormatInfoSelector\_SCF1WRWG12p, [279](#)
- PixelFormatInfoSelector\_SCF1WRWG14, [279](#)
- PixelFormatInfoSelector\_SCF1WRWG16, [279](#)
- PixelFormatInfoSelector\_SCF1WRWG8, [279](#)
- PixelFormatInfoSelector\_YCbCr10\_CbYCr, [279](#)
- PixelFormatInfoSelector\_YCbCr10p\_CbYCr, [279](#)

- PixelFormatInfoSelector\_YCbCr12\_CbYCr, [280](#)  
PixelFormatInfoSelector\_YCbCr12p\_CbYCr, [280](#)  
PixelFormatInfoSelector\_YCbCr411\_8, [280](#)  
PixelFormatInfoSelector\_YCbCr411\_8\_CbYYCrYY, [280](#)  
PixelFormatInfoSelector\_YCbCr422\_10, [280](#)  
PixelFormatInfoSelector\_YCbCr422\_10\_CbYCrY, [280](#)  
PixelFormatInfoSelector\_YCbCr422\_10p, [280](#)  
PixelFormatInfoSelector\_YCbCr422\_10p\_CbYCrY, [280](#)  
PixelFormatInfoSelector\_YCbCr422\_12, [280](#)  
PixelFormatInfoSelector\_YCbCr422\_12\_CbYCrY, [280](#)  
PixelFormatInfoSelector\_YCbCr422\_12p, [280](#)  
PixelFormatInfoSelector\_YCbCr422\_12p\_CbYCrY, [280](#)  
PixelFormatInfoSelector\_YCbCr422\_8, [280](#)  
PixelFormatInfoSelector\_YCbCr422\_8\_CbYCrY, [280](#)  
PixelFormatInfoSelector\_YCbCr601\_10\_CbYCr, [280](#)  
PixelFormatInfoSelector\_YCbCr601\_10p\_CbYCr, [280](#)  
PixelFormatInfoSelector\_YCbCr601\_12\_CbYCr, [280](#)  
PixelFormatInfoSelector\_YCbCr601\_12p\_CbYCr, [280](#)  
PixelFormatInfoSelector\_YCbCr601\_411\_8\_CbYYCrYY, [280](#)  
PixelFormatInfoSelector\_YCbCr601\_422\_10, [280](#)  
PixelFormatInfoSelector\_YCbCr601\_422\_10\_CbYCrY, [280](#)  
PixelFormatInfoSelector\_YCbCr601\_422\_10p, [280](#)  
PixelFormatInfoSelector\_YCbCr601\_422\_10p\_CbYCrY, [280](#)  
PixelFormatInfoSelector\_YCbCr601\_422\_12, [280](#)  
PixelFormatInfoSelector\_YCbCr601\_422\_12\_CbYCrY, [280](#)  
PixelFormatInfoSelector\_YCbCr601\_422\_12p, [280](#)  
PixelFormatInfoSelector\_YCbCr601\_422\_12p\_CbYCrY, [280](#)  
PixelFormatInfoSelector\_YCbCr601\_422\_8, [280](#)  
PixelFormatInfoSelector\_YCbCr601\_422\_8\_CbYCrY, [280](#)  
PixelFormatInfoSelector\_YCbCr601\_8\_CbYCr, [280](#)  
PixelFormatInfoSelector\_YCbCr709\_10\_CbYCr, [280](#)  
PixelFormatInfoSelector\_YCbCr709\_10p\_CbYCr, [280](#)  
PixelFormatInfoSelector\_YCbCr709\_12\_CbYCr, [280](#)  
PixelFormatInfoSelector\_YCbCr709\_12p\_CbYCr, [280](#)  
PixelFormatInfoSelector\_YCbCr709\_411\_8\_CbYYCrYY, [280](#)  
PixelFormatInfoSelector\_YCbCr709\_422\_10, [280](#)  
PixelFormatInfoSelector\_YCbCr709\_422\_10\_CbYCrY, [280](#)  
PixelFormatInfoSelector\_YCbCr709\_422\_10p, [280](#)  
PixelFormatInfoSelector\_YCbCr709\_422\_10p\_CbYCrY, [281](#)  
PixelFormatInfoSelector\_YCbCr709\_422\_12, [281](#)  
PixelFormatInfoSelector\_YCbCr709\_422\_12\_CbYCrY, [281](#)  
PixelFormatInfoSelector\_YCbCr709\_422\_12p, [281](#)  
PixelFormatInfoSelector\_YCbCr709\_422\_12p\_CbYCrY, [281](#)  
PixelFormatInfoSelector\_YCbCr709\_422\_8, [280](#)  
PixelFormatInfoSelector\_YCbCr709\_422\_8\_CbYCrY, [280](#)  
PixelFormatInfoSelector\_YCbCr709\_8\_CbYCr, [280](#)  
PixelFormatInfoSelector\_YCbCr8, [279](#)  
PixelFormatInfoSelector\_YCbCr8\_CbYCr, [279](#)  
PixelFormatInfoSelector\_YUV411\_8\_UYYVYY, [281](#)  
PixelFormatInfoSelector\_YUV422\_8, [281](#)  
PixelFormatInfoSelector\_YUV422\_8\_UYVY, [281](#)  
PixelFormatInfoSelector\_YUV8\_UYV, [281](#)  
PixelSize\_Bpp1, [281](#)  
PixelSize\_Bpp10, [281](#)  
PixelSize\_Bpp12, [281](#)  
PixelSize\_Bpp14, [281](#)  
PixelSize\_Bpp16, [281](#)  
PixelSize\_Bpp2, [281](#)  
PixelSize\_Bpp20, [281](#)  
PixelSize\_Bpp24, [281](#)  
PixelSize\_Bpp30, [281](#)  
PixelSize\_Bpp32, [281](#)  
PixelSize\_Bpp36, [282](#)  
PixelSize\_Bpp4, [281](#)  
PixelSize\_Bpp48, [282](#)  
PixelSize\_Bpp64, [282](#)  
PixelSize\_Bpp8, [281](#)  
PixelSize\_Bpp96, [282](#)  
RegionDestination\_Stream0, [282](#)  
RegionDestination\_Stream1, [282](#)  
RegionDestination\_Stream2, [282](#)  
RegionMode\_Off, [282](#)  
RegionMode\_On, [282](#)  
RegionSelector\_All, [283](#)  
RegionSelector\_Region0, [283](#)  
RegionSelector\_Region1, [283](#)  
RegionSelector\_Region2, [283](#)  
RgbTransformLightSource\_Cloudy6500K, [283](#)  
RgbTransformLightSource\_CoolFluorescent4000K, [283](#)  
RgbTransformLightSource\_Custom, [283](#)  
RgbTransformLightSource\_Daylight5000K, [283](#)  
RgbTransformLightSource\_General, [283](#)  
RgbTransformLightSource\_Shade8000K, [283](#)  
RgbTransformLightSource\_Tungsten2800K, [283](#)  
RgbTransformLightSource\_WarmFluorescent3000K, [283](#)

- Scan3dCoordinateReferenceSelector\_RotationX, [284](#)
- Scan3dCoordinateReferenceSelector\_RotationY, [284](#)
- Scan3dCoordinateReferenceSelector\_RotationZ, [284](#)
- Scan3dCoordinateReferenceSelector\_TranslationX, [284](#)
- Scan3dCoordinateReferenceSelector\_TranslationY, [284](#)
- Scan3dCoordinateReferenceSelector\_TranslationZ, [284](#)
- Scan3dCoordinateSelector\_CoordinateA, [284](#)
- Scan3dCoordinateSelector\_CoordinateB, [284](#)
- Scan3dCoordinateSelector\_CoordinateC, [284](#)
- Scan3dCoordinateSystem\_Cartesian, [284](#)
- Scan3dCoordinateSystem\_Cylindrical, [284](#)
- Scan3dCoordinateSystem\_Spherical, [284](#)
- Scan3dCoordinateSystemReference\_Anchor, [285](#)
- Scan3dCoordinateSystemReference\_Transformed, [285](#)
- Scan3dCoordinateTransformSelector\_RotationX, [285](#)
- Scan3dCoordinateTransformSelector\_RotationY, [285](#)
- Scan3dCoordinateTransformSelector\_RotationZ, [285](#)
- Scan3dCoordinateTransformSelector\_TranslationX, [285](#)
- Scan3dCoordinateTransformSelector\_TranslationY, [285](#)
- Scan3dCoordinateTransformSelector\_TranslationZ, [285](#)
- Scan3dDistanceUnit\_Inch, [285](#)
- Scan3dDistanceUnit\_Millimeter, [285](#)
- Scan3dOutputMode\_CalibratedABC\_Grid, [286](#)
- Scan3dOutputMode\_CalibratedABC\_PointCloud, [286](#)
- Scan3dOutputMode\_CalibratedAC, [286](#)
- Scan3dOutputMode\_CalibratedAC\_Linescan, [286](#)
- Scan3dOutputMode\_CalibratedC, [286](#)
- Scan3dOutputMode\_CalibratedC\_Linescan, [286](#)
- Scan3dOutputMode\_DisparityC, [286](#)
- Scan3dOutputMode\_DisparityC\_Linescan, [286](#)
- Scan3dOutputMode\_RectifiedC, [286](#)
- Scan3dOutputMode\_RectifiedC\_Linescan, [286](#)
- Scan3dOutputMode\_UncalibratedC, [286](#)
- SensorDigitizationTaps\_Eight, [287](#)
- SensorDigitizationTaps\_Four, [287](#)
- SensorDigitizationTaps\_One, [286](#)
- SensorDigitizationTaps\_Ten, [287](#)
- SensorDigitizationTaps\_Three, [287](#)
- SensorDigitizationTaps\_Two, [287](#)
- SensorShutterMode\_Global, [287](#)
- SensorShutterMode\_GlobalReset, [287](#)
- SensorShutterMode\_Rolling, [287](#)
- SensorTaps\_Eight, [287](#)
- SensorTaps\_Four, [287](#)
- SensorTaps\_One, [287](#)
- SensorTaps\_Ten, [287](#)
- SensorTaps\_Three, [287](#)
- SensorTaps\_Two, [287](#)
- SequencerConfigurationMode\_Off, [288](#)
- SequencerConfigurationMode\_On, [288](#)
- SequencerConfigurationValid\_No, [288](#)
- SequencerConfigurationValid\_Yes, [288](#)
- SequencerMode\_Off, [288](#)
- SequencerMode\_On, [288](#)
- SequencerSetValid\_No, [289](#)
- SequencerSetValid\_Yes, [289](#)
- SequencerTriggerActivation\_AnyEdge, [289](#)
- SequencerTriggerActivation\_FallingEdge, [289](#)
- SequencerTriggerActivation\_LevelHigh, [289](#)
- SequencerTriggerActivation\_LevelLow, [289](#)
- SequencerTriggerActivation\_RisingEdge, [289](#)
- SequencerTriggerSource\_FrameStart, [289](#)
- SequencerTriggerSource\_Off, [289](#)
- SerialPortBaudRate\_Baud115200, [290](#)
- SerialPortBaudRate\_Baud1200, [290](#)
- SerialPortBaudRate\_Baud14400, [290](#)
- SerialPortBaudRate\_Baud19200, [290](#)
- SerialPortBaudRate\_Baud230400, [290](#)
- SerialPortBaudRate\_Baud2400, [290](#)
- SerialPortBaudRate\_Baud300, [290](#)
- SerialPortBaudRate\_Baud38400, [290](#)
- SerialPortBaudRate\_Baud460800, [290](#)
- SerialPortBaudRate\_Baud4800, [290](#)
- SerialPortBaudRate\_Baud57600, [290](#)
- SerialPortBaudRate\_Baud600, [290](#)
- SerialPortBaudRate\_Baud921600, [290](#)
- SerialPortBaudRate\_Baud9600, [290](#)
- SerialPortParity\_Even, [290](#)
- SerialPortParity\_Mark, [290](#)
- SerialPortParity\_None, [290](#)
- SerialPortParity\_Odd, [290](#)
- SerialPortParity\_Space, [290](#)
- SerialPortSelector\_SerialPort0, [290](#)
- SerialPortSource\_Line0, [291](#)
- SerialPortSource\_Line1, [291](#)
- SerialPortSource\_Line2, [291](#)
- SerialPortSource\_Line3, [291](#)
- SerialPortSource\_Off, [291](#)
- SerialPortStopBits\_Bits1, [291](#)
- SerialPortStopBits\_Bits1AndAHalf, [291](#)
- SerialPortStopBits\_Bits2, [291](#)
- SoftwareSignalSelector\_SoftwareSignal0, [291](#)
- SoftwareSignalSelector\_SoftwareSignal1, [291](#)
- SoftwareSignalSelector\_SoftwareSignal2, [291](#)
- SourceSelector\_All, [292](#)
- SourceSelector\_Source0, [292](#)
- SourceSelector\_Source1, [292](#)
- SourceSelector\_Source2, [292](#)
- TestPattern\_Increment, [292](#)
- TestPattern\_Off, [292](#)
- TestPattern\_SensorTestPattern, [292](#)
- TestPatternGeneratorSelector\_PipelineStart, [292](#)



- TestPatternGeneratorSelector\_Sensor, [292](#)
- TimerSelector\_Timer0, [293](#)
- TimerSelector\_Timer1, [293](#)
- TimerSelector\_Timer2, [293](#)
- TimerStatus\_TimerActive, [293](#)
- TimerStatus\_TimerCompleted, [293](#)
- TimerStatus\_TimerIdle, [293](#)
- TimerStatus\_TimerTriggerWait, [293](#)
- TimerTriggerActivation\_AnyEdge, [293](#)
- TimerTriggerActivation\_FallingEdge, [293](#)
- TimerTriggerActivation\_LevelHigh, [293](#)
- TimerTriggerActivation\_LevelLow, [293](#)
- TimerTriggerActivation\_RisingEdge, [293](#)
- TimerTriggerSource\_AcquisitionEnd, [294](#)
- TimerTriggerSource\_AcquisitionStart, [294](#)
- TimerTriggerSource\_AcquisitionTrigger, [294](#)
- TimerTriggerSource\_Action0, [295](#)
- TimerTriggerSource\_Action1, [295](#)
- TimerTriggerSource\_Action2, [295](#)
- TimerTriggerSource\_Counter0End, [294](#)
- TimerTriggerSource\_Counter0Start, [294](#)
- TimerTriggerSource\_Counter1End, [294](#)
- TimerTriggerSource\_Counter1Start, [294](#)
- TimerTriggerSource\_Counter2End, [294](#)
- TimerTriggerSource\_Counter2Start, [294](#)
- TimerTriggerSource\_Encoder0, [295](#)
- TimerTriggerSource\_Encoder1, [295](#)
- TimerTriggerSource\_Encoder2, [295](#)
- TimerTriggerSource\_ExposureEnd, [294](#)
- TimerTriggerSource\_ExposureStart, [294](#)
- TimerTriggerSource\_FrameBurstEnd, [294](#)
- TimerTriggerSource\_FrameBurstStart, [294](#)
- TimerTriggerSource\_FrameEnd, [294](#)
- TimerTriggerSource\_FrameStart, [294](#)
- TimerTriggerSource\_FrameTrigger, [294](#)
- TimerTriggerSource\_Line0, [294](#)
- TimerTriggerSource\_Line1, [294](#)
- TimerTriggerSource\_Line2, [294](#)
- TimerTriggerSource\_LineEnd, [294](#)
- TimerTriggerSource\_LineStart, [294](#)
- TimerTriggerSource\_LineTrigger, [294](#)
- TimerTriggerSource\_LinkTrigger0, [295](#)
- TimerTriggerSource\_LinkTrigger1, [295](#)
- TimerTriggerSource\_LinkTrigger2, [295](#)
- TimerTriggerSource\_Off, [294](#)
- TimerTriggerSource\_SoftwareSignal0, [295](#)
- TimerTriggerSource\_SoftwareSignal1, [295](#)
- TimerTriggerSource\_SoftwareSignal2, [295](#)
- TimerTriggerSource\_Timer0End, [294](#)
- TimerTriggerSource\_Timer0Start, [294](#)
- TimerTriggerSource\_Timer1End, [294](#)
- TimerTriggerSource\_Timer1Start, [294](#)
- TimerTriggerSource\_Timer2End, [294](#)
- TimerTriggerSource\_Timer2Start, [294](#)
- TimerTriggerSource\_UserOutput0, [294](#)
- TimerTriggerSource\_UserOutput1, [294](#)
- TimerTriggerSource\_UserOutput2, [294](#)
- TransferComponentSelector\_All, [295](#)
- TransferComponentSelector\_Blue, [295](#)
- TransferComponentSelector\_Green, [295](#)
- TransferComponentSelector\_Red, [295](#)
- TransferControlMode\_Automatic, [296](#)
- TransferControlMode\_Basic, [296](#)
- TransferControlMode\_UserControlled, [296](#)
- TransferOperationMode\_Continuous, [296](#)
- TransferOperationMode\_MultiBlock, [296](#)
- TransferQueueMode\_FirstInFirstOut, [296](#)
- TransferSelector\_All, [296](#)
- TransferSelector\_Stream0, [296](#)
- TransferSelector\_Stream1, [296](#)
- TransferSelector\_Stream2, [296](#)
- TransferStatusSelector\_Paused, [297](#)
- TransferStatusSelector\_QueueOverflow, [297](#)
- TransferStatusSelector\_Stopped, [297](#)
- TransferStatusSelector\_Stopping, [297](#)
- TransferStatusSelector\_Streaming, [297](#)
- TransferTriggerActivation\_AnyEdge, [297](#)
- TransferTriggerActivation\_FallingEdge, [297](#)
- TransferTriggerActivation\_LevelHigh, [297](#)
- TransferTriggerActivation\_LevelLow, [297](#)
- TransferTriggerActivation\_RisingEdge, [297](#)
- TransferTriggerMode\_Off, [298](#)
- TransferTriggerMode\_On, [298](#)
- TransferTriggerSelector\_TransferAbort, [298](#)
- TransferTriggerSelector\_TransferActive, [298](#)
- TransferTriggerSelector\_TransferBurstStart, [298](#)
- TransferTriggerSelector\_TransferBurstStop, [298](#)
- TransferTriggerSelector\_TransferPause, [298](#)
- TransferTriggerSelector\_TransferResume, [298](#)
- TransferTriggerSelector\_TransferStart, [298](#)
- TransferTriggerSelector\_TransferStop, [298](#)
- TransferTriggerSource\_Action0, [299](#)
- TransferTriggerSource\_Action1, [299](#)
- TransferTriggerSource\_Action2, [299](#)
- TransferTriggerSource\_Counter0End, [299](#)
- TransferTriggerSource\_Counter0Start, [298](#)
- TransferTriggerSource\_Counter1End, [299](#)
- TransferTriggerSource\_Counter1Start, [298](#)
- TransferTriggerSource\_Counter2End, [299](#)
- TransferTriggerSource\_Counter2Start, [299](#)
- TransferTriggerSource\_Line0, [298](#)
- TransferTriggerSource\_Line1, [298](#)
- TransferTriggerSource\_Line2, [298](#)
- TransferTriggerSource\_SoftwareSignal0, [299](#)
- TransferTriggerSource\_SoftwareSignal1, [299](#)
- TransferTriggerSource\_SoftwareSignal2, [299](#)
- TransferTriggerSource\_Timer0End, [299](#)
- TransferTriggerSource\_Timer0Start, [299](#)
- TransferTriggerSource\_Timer1End, [299](#)
- TransferTriggerSource\_Timer1Start, [299](#)
- TransferTriggerSource\_Timer2End, [299](#)
- TransferTriggerSource\_Timer2Start, [299](#)
- TriggerActivation\_AnyEdge, [299](#)
- TriggerActivation\_FallingEdge, [299](#)
- TriggerActivation\_LevelHigh, [299](#)
- TriggerActivation\_LevelLow, [299](#)

- TriggerActivation\_RisingEdge, [299](#)
- TriggerMode\_Off, [300](#)
- TriggerMode\_On, [300](#)
- TriggerOverlap\_Off, [300](#)
- TriggerOverlap\_PreviousFrame, [300](#)
- TriggerOverlap\_ReadOut, [300](#)
- TriggerSelector\_AcquisitionStart, [300](#)
- TriggerSelector\_FrameBurstStart, [300](#)
- TriggerSelector\_FrameStart, [300](#)
- TriggerSource\_Action0, [301](#)
- TriggerSource\_Counter0End, [301](#)
- TriggerSource\_Counter0Start, [301](#)
- TriggerSource\_Counter1End, [301](#)
- TriggerSource\_Counter1Start, [301](#)
- TriggerSource\_Line0, [301](#)
- TriggerSource\_Line1, [301](#)
- TriggerSource\_Line2, [301](#)
- TriggerSource\_Line3, [301](#)
- TriggerSource\_LogicBlock0, [301](#)
- TriggerSource\_LogicBlock1, [301](#)
- TriggerSource\_Software, [301](#)
- TriggerSource\_UserOutput0, [301](#)
- TriggerSource\_UserOutput1, [301](#)
- TriggerSource\_UserOutput2, [301](#)
- TriggerSource\_UserOutput3, [301](#)
- UNKNOWN\_PIXELFORMAT, [275](#)
- UserOutputSelector\_UserOutput0, [301](#)
- UserOutputSelector\_UserOutput1, [301](#)
- UserOutputSelector\_UserOutput2, [301](#)
- UserOutputSelector\_UserOutput3, [301](#)
- UserSetDefault\_Default, [302](#)
- UserSetDefault\_UserSet0, [302](#)
- UserSetDefault\_UserSet1, [302](#)
- UserSetSelector\_Default, [302](#)
- UserSetSelector\_UserSet0, [302](#)
- UserSetSelector\_UserSet1, [302](#)
- WhiteClipSelector\_All, [302](#)
- WhiteClipSelector\_Blue, [302](#)
- WhiteClipSelector\_Green, [302](#)
- WhiteClipSelector\_Red, [302](#)
- WhiteClipSelector\_Tap1, [302](#)
- WhiteClipSelector\_Tap2, [302](#)
- WhiteClipSelector\_U, [302](#)
- WhiteClipSelector\_V, [302](#)
- WhiteClipSelector\_Y, [302](#)
- CameraList Access, [16](#)
- CategoryNode
  - SpinnakerGenApiDefsC.h, [461](#)
- CCITTFA3
  - SpinnakerDefsC.h, [413](#)
- CCITTFAX4
  - SpinnakerDefsC.h, [413](#)
- Chunk data access, [24](#)
- Chunk Data Structures, [9](#)
- ChunkBlackLevel
  - \_quickSpin, [71](#)
- ChunkBlackLevelSelector
  - \_quickSpin, [72](#)
- ChunkBlackLevelSelector\_All
  - CameraDefsC.h, [226](#)
- ChunkCounterSelector
  - \_quickSpin, [72](#)
- ChunkCounterSelector\_Counter0
  - CameraDefsC.h, [226](#)
- ChunkCounterSelector\_Counter1
  - CameraDefsC.h, [226](#)
- ChunkCounterSelector\_Counter2
  - CameraDefsC.h, [226](#)
- ChunkCounterValue
  - \_quickSpin, [72](#)
- ChunkCRC
  - \_quickSpin, [72](#)
- ChunkEnable
  - \_quickSpin, [72](#)
- ChunkEncoderSelector
  - \_quickSpin, [72](#)
- ChunkEncoderSelector\_Encoder0
  - CameraDefsC.h, [226](#)
- ChunkEncoderSelector\_Encoder1
  - CameraDefsC.h, [226](#)
- ChunkEncoderSelector\_Encoder2
  - CameraDefsC.h, [226](#)
- ChunkEncoderStatus
  - \_quickSpin, [72](#)
- ChunkEncoderStatus\_EncoderDown
  - CameraDefsC.h, [227](#)
- ChunkEncoderStatus\_EncoderIdle
  - CameraDefsC.h, [227](#)
- ChunkEncoderStatus\_EncoderStatic
  - CameraDefsC.h, [227](#)
- ChunkEncoderStatus\_EncoderUp
  - CameraDefsC.h, [227](#)
- ChunkEncoderValue
  - \_quickSpin, [72](#)
- ChunkExposureEndLineStatusAll
  - \_quickSpin, [73](#)
- ChunkExposureTime
  - \_quickSpin, [73](#)
- ChunkExposureTimeSelector
  - \_quickSpin, [73](#)
- ChunkExposureTimeSelector\_Blue
  - CameraDefsC.h, [227](#)
- ChunkExposureTimeSelector\_Common
  - CameraDefsC.h, [227](#)
- ChunkExposureTimeSelector\_Cyan
  - CameraDefsC.h, [227](#)
- ChunkExposureTimeSelector\_Green
  - CameraDefsC.h, [227](#)
- ChunkExposureTimeSelector\_Infrared
  - CameraDefsC.h, [227](#)
- ChunkExposureTimeSelector\_Magenta
  - CameraDefsC.h, [227](#)
- ChunkExposureTimeSelector\_Red
  - CameraDefsC.h, [227](#)
- ChunkExposureTimeSelector\_Stage1
  - CameraDefsC.h, [227](#)

- ChunkExposureTimeSelector\_Stage2
  - CameraDefsC.h, [227](#)
- ChunkExposureTimeSelector\_Ultraviolet
  - CameraDefsC.h, [227](#)
- ChunkExposureTimeSelector\_Yellow
  - CameraDefsC.h, [227](#)
- ChunkFrameID
  - \_quickSpin, [73](#)
- ChunkGain
  - \_quickSpin, [73](#)
- ChunkGainSelector
  - \_quickSpin, [73](#)
- ChunkGainSelector\_All
  - CameraDefsC.h, [227](#)
- ChunkGainSelector\_Blue
  - CameraDefsC.h, [227](#)
- ChunkGainSelector\_Green
  - CameraDefsC.h, [227](#)
- ChunkGainSelector\_Red
  - CameraDefsC.h, [227](#)
- ChunkHeight
  - \_quickSpin, [73](#)
- ChunkImage
  - \_quickSpin, [73](#)
- ChunkImageComponent
  - \_quickSpin, [74](#)
- ChunkImageComponent\_Color
  - CameraDefsC.h, [228](#)
- ChunkImageComponent\_Confidence
  - CameraDefsC.h, [228](#)
- ChunkImageComponent\_Disparity
  - CameraDefsC.h, [228](#)
- ChunkImageComponent\_Infrared
  - CameraDefsC.h, [228](#)
- ChunkImageComponent\_Intensity
  - CameraDefsC.h, [228](#)
- ChunkImageComponent\_Range
  - CameraDefsC.h, [228](#)
- ChunkImageComponent\_Scatter
  - CameraDefsC.h, [228](#)
- ChunkImageComponent\_Ultraviolet
  - CameraDefsC.h, [228](#)
- ChunkInferenceBoundingBoxResult
  - \_quickSpin, [74](#)
- ChunkInferenceConfidence
  - \_quickSpin, [74](#)
- ChunkInferenceFrameId
  - \_quickSpin, [74](#)
- ChunkInferenceResult
  - \_quickSpin, [74](#)
- ChunkLinePitch
  - \_quickSpin, [74](#)
- ChunkLineStatusAll
  - \_quickSpin, [74](#)
- ChunkModeActive
  - \_quickSpin, [74](#)
- ChunkOffsetX
  - \_quickSpin, [75](#)
- ChunkOffsetY
  - \_quickSpin, [75](#)
- ChunkPartSelector
  - \_quickSpin, [75](#)
- ChunkPixelDynamicRangeMax
  - \_quickSpin, [75](#)
- ChunkPixelDynamicRangeMin
  - \_quickSpin, [75](#)
- ChunkPixelFormat
  - \_quickSpin, [75](#)
- ChunkPixelFormat\_BayerBG8
  - CameraDefsC.h, [228](#)
- ChunkPixelFormat\_BayerGB8
  - CameraDefsC.h, [228](#)
- ChunkPixelFormat\_BayerGR8
  - CameraDefsC.h, [228](#)
- ChunkPixelFormat\_BayerRG8
  - CameraDefsC.h, [228](#)
- ChunkPixelFormat\_Mono12Packed
  - CameraDefsC.h, [228](#)
- ChunkPixelFormat\_Mono16
  - CameraDefsC.h, [228](#)
- ChunkPixelFormat\_Mono8
  - CameraDefsC.h, [228](#)
- ChunkPixelFormat\_RGB8Packed
  - CameraDefsC.h, [228](#)
- ChunkPixelFormat\_YCbCr601\_422\_8\_CbYCrY
  - CameraDefsC.h, [228](#)
- ChunkPixelFormat\_YUV422Packed
  - CameraDefsC.h, [228](#)
- ChunkRegionID
  - \_quickSpin, [75](#)
- ChunkRegionID\_Region0
  - CameraDefsC.h, [229](#)
- ChunkRegionID\_Region1
  - CameraDefsC.h, [229](#)
- ChunkRegionID\_Region2
  - CameraDefsC.h, [229](#)
- ChunkScan3dAxisMax
  - \_quickSpin, [75](#)
- ChunkScan3dAxisMin
  - \_quickSpin, [76](#)
- ChunkScan3dCoordinateOffset
  - \_quickSpin, [76](#)
- ChunkScan3dCoordinateReferenceSelector
  - \_quickSpin, [76](#)
- ChunkScan3dCoordinateReferenceSelector\_RotationX
  - CameraDefsC.h, [229](#)
- ChunkScan3dCoordinateReferenceSelector\_RotationY
  - CameraDefsC.h, [229](#)
- ChunkScan3dCoordinateReferenceSelector\_RotationZ
  - CameraDefsC.h, [229](#)
- ChunkScan3dCoordinateReferenceSelector\_TranslationX
  - CameraDefsC.h, [229](#)
- ChunkScan3dCoordinateReferenceSelector\_TranslationY
  - CameraDefsC.h, [229](#)
- ChunkScan3dCoordinateReferenceSelector\_TranslationZ
  - CameraDefsC.h, [229](#)

- ChunkScan3dCoordinateReferenceValue
  - \_quickSpin, [76](#)
- ChunkScan3dCoordinateScale
  - \_quickSpin, [76](#)
- ChunkScan3dCoordinateSelector
  - \_quickSpin, [76](#)
- ChunkScan3dCoordinateSelector\_CoordinateA
  - CameraDefsC.h, [229](#)
- ChunkScan3dCoordinateSelector\_CoordinateB
  - CameraDefsC.h, [229](#)
- ChunkScan3dCoordinateSelector\_CoordinateC
  - CameraDefsC.h, [229](#)
- ChunkScan3dCoordinateSystem
  - \_quickSpin, [76](#)
- ChunkScan3dCoordinateSystem\_Cartesian
  - CameraDefsC.h, [230](#)
- ChunkScan3dCoordinateSystem\_Cylindrical
  - CameraDefsC.h, [230](#)
- ChunkScan3dCoordinateSystem\_Spherical
  - CameraDefsC.h, [230](#)
- ChunkScan3dCoordinateSystemReference
  - \_quickSpin, [76](#)
- ChunkScan3dCoordinateSystemReference\_Anchor
  - CameraDefsC.h, [230](#)
- ChunkScan3dCoordinateSystemReference\_Transformed
  - CameraDefsC.h, [230](#)
- ChunkScan3dCoordinateTransformSelector
  - \_quickSpin, [77](#)
- ChunkScan3dCoordinateTransformSelector\_RotationX
  - CameraDefsC.h, [230](#)
- ChunkScan3dCoordinateTransformSelector\_RotationY
  - CameraDefsC.h, [230](#)
- ChunkScan3dCoordinateTransformSelector\_RotationZ
  - CameraDefsC.h, [230](#)
- ChunkScan3dCoordinateTransformSelector\_TranslationX
  - CameraDefsC.h, [230](#)
- ChunkScan3dCoordinateTransformSelector\_TranslationY
  - CameraDefsC.h, [230](#)
- ChunkScan3dCoordinateTransformSelector\_TranslationZ
  - CameraDefsC.h, [230](#)
- ChunkScan3dDistanceUnit
  - \_quickSpin, [77](#)
- ChunkScan3dDistanceUnit\_Inch
  - CameraDefsC.h, [231](#)
- ChunkScan3dDistanceUnit\_Millimeter
  - CameraDefsC.h, [231](#)
- ChunkScan3dInvalidDataFlag
  - \_quickSpin, [77](#)
- ChunkScan3dInvalidDataValue
  - \_quickSpin, [77](#)
- ChunkScan3dOutputMode
  - \_quickSpin, [77](#)
- ChunkScan3dOutputMode\_CalibratedABC\_Grid
  - CameraDefsC.h, [231](#)
- ChunkScan3dOutputMode\_CalibratedABC\_PointCloud
  - CameraDefsC.h, [231](#)
- ChunkScan3dOutputMode\_CalibratedAC
  - CameraDefsC.h, [231](#)
- ChunkScan3dOutputMode\_CalibratedAC\_Linescan
  - CameraDefsC.h, [231](#)
- ChunkScan3dOutputMode\_CalibratedC
  - CameraDefsC.h, [231](#)
- ChunkScan3dOutputMode\_CalibratedC\_Linescan
  - CameraDefsC.h, [231](#)
- ChunkScan3dOutputMode\_DisparityC
  - CameraDefsC.h, [232](#)
- ChunkScan3dOutputMode\_DisparityC\_Linescan
  - CameraDefsC.h, [232](#)
- ChunkScan3dOutputMode\_RectifiedC
  - CameraDefsC.h, [231](#)
- ChunkScan3dOutputMode\_RectifiedC\_Linescan
  - CameraDefsC.h, [232](#)
- ChunkScan3dOutputMode\_UncalibratedC
  - CameraDefsC.h, [231](#)
- ChunkScan3dTransformValue
  - \_quickSpin, [77](#)
- ChunkScanLineSelector
  - \_quickSpin, [77](#)
- ChunkSelector
  - \_quickSpin, [77](#)
- ChunkSelector\_BlackLevel
  - CameraDefsC.h, [232](#)
- ChunkSelector\_CRC
  - CameraDefsC.h, [232](#)
- ChunkSelector\_ExposureEndLineStatusAll
  - CameraDefsC.h, [232](#)
- ChunkSelector\_ExposureTime
  - CameraDefsC.h, [232](#)
- ChunkSelector\_FrameID
  - CameraDefsC.h, [232](#)
- ChunkSelector\_Gain
  - CameraDefsC.h, [232](#)
- ChunkSelector\_Height
  - CameraDefsC.h, [232](#)
- ChunkSelector\_Image
  - CameraDefsC.h, [232](#)
- ChunkSelector\_OffsetX
  - CameraDefsC.h, [232](#)
- ChunkSelector\_OffsetY
  - CameraDefsC.h, [232](#)
- ChunkSelector\_PixelFormat
  - CameraDefsC.h, [232](#)
- ChunkSelector\_SequencerSetActive
  - CameraDefsC.h, [232](#)
- ChunkSelector\_SerialData
  - CameraDefsC.h, [232](#)
- ChunkSelector\_Timestamp
  - CameraDefsC.h, [232](#)
- ChunkSelector\_Width
  - CameraDefsC.h, [232](#)
- ChunkSequencerSetActive
  - \_quickSpin, [78](#)
- ChunkSerialData
  - \_quickSpin, [78](#)
- ChunkSerialDataLength
  - \_quickSpin, [78](#)



- ChunkSerialReceiveOverflow
  - \_quickSpin, [78](#)
- ChunkSourceID
  - \_quickSpin, [78](#)
- ChunkSourceID\_Source0
  - CameraDefsC.h, [233](#)
- ChunkSourceID\_Source1
  - CameraDefsC.h, [233](#)
- ChunkSourceID\_Source2
  - CameraDefsC.h, [233](#)
- ChunkStreamChannelID
  - \_quickSpin, [78](#)
- ChunkTimerSelector
  - \_quickSpin, [78](#)
- ChunkTimerSelector\_Timer0
  - CameraDefsC.h, [233](#)
- ChunkTimerSelector\_Timer1
  - CameraDefsC.h, [233](#)
- ChunkTimerSelector\_Timer2
  - CameraDefsC.h, [233](#)
- ChunkTimerValue
  - \_quickSpin, [78](#)
- ChunkTimestamp
  - \_quickSpin, [79](#)
- ChunkTimestampLatchValue
  - \_quickSpin, [79](#)
- ChunkTransferBlockID
  - \_quickSpin, [79](#)
- ChunkTransferQueueCurrentBlockCount
  - \_quickSpin, [79](#)
- ChunkTransferStreamID
  - \_quickSpin, [79](#)
- ChunkTransferStreamID\_Stream0
  - CameraDefsC.h, [233](#)
- ChunkTransferStreamID\_Stream1
  - CameraDefsC.h, [233](#)
- ChunkTransferStreamID\_Stream2
  - CameraDefsC.h, [233](#)
- ChunkTransferStreamID\_Stream3
  - CameraDefsC.h, [233](#)
- ChunkWidth
  - \_quickSpin, [79](#)
- CL
  - SpinnakerGenApiDefsC.h, [462](#)
- CIConfiguration
  - \_quickSpin, [79](#)
- CIConfiguration\_Base
  - CameraDefsC.h, [234](#)
- CIConfiguration\_DualBase
  - CameraDefsC.h, [234](#)
- CIConfiguration\_EightyBit
  - CameraDefsC.h, [234](#)
- CIConfiguration\_Full
  - CameraDefsC.h, [234](#)
- CIConfiguration\_Medium
  - CameraDefsC.h, [234](#)
- CITimeSlotsCount
  - \_quickSpin, [79](#)
- CITimeSlotsCount\_One
  - CameraDefsC.h, [234](#)
- CITimeSlotsCount\_Three
  - CameraDefsC.h, [234](#)
- CITimeSlotsCount\_Two
  - CameraDefsC.h, [234](#)
- ColorTransformationEnable
  - \_quickSpin, [80](#)
- ColorTransformationSelector
  - \_quickSpin, [80](#)
- ColorTransformationSelector\_RGBtoRGB
  - CameraDefsC.h, [234](#)
- ColorTransformationSelector\_RGBtoYUV
  - CameraDefsC.h, [234](#)
- ColorTransformationValue
  - \_quickSpin, [80](#)
- ColorTransformationValueSelector
  - \_quickSpin, [80](#)
- ColorTransformationValueSelector\_Gain00
  - CameraDefsC.h, [235](#)
- ColorTransformationValueSelector\_Gain01
  - CameraDefsC.h, [235](#)
- ColorTransformationValueSelector\_Gain02
  - CameraDefsC.h, [235](#)
- ColorTransformationValueSelector\_Gain10
  - CameraDefsC.h, [235](#)
- ColorTransformationValueSelector\_Gain11
  - CameraDefsC.h, [235](#)
- ColorTransformationValueSelector\_Gain12
  - CameraDefsC.h, [235](#)
- ColorTransformationValueSelector\_Gain20
  - CameraDefsC.h, [235](#)
- ColorTransformationValueSelector\_Gain21
  - CameraDefsC.h, [235](#)
- ColorTransformationValueSelector\_Gain22
  - CameraDefsC.h, [235](#)
- ColorTransformationValueSelector\_Offset0
  - CameraDefsC.h, [235](#)
- ColorTransformationValueSelector\_Offset1
  - CameraDefsC.h, [235](#)
- ColorTransformationValueSelector\_Offset2
  - CameraDefsC.h, [235](#)
- CommandNode
  - SpinnakerGenApiDefsC.h, [461](#)
- compression
  - \_spinTIFFOption, [186](#)
- compressionLevel
  - \_spinPNGOption, [184](#)
- CompressionMethod
  - SpinnakerDefsC.h, [413](#)
- CompressionRatio
  - \_quickSpin, [80](#)
- CounterDelay
  - \_quickSpin, [80](#)
- CounterDuration
  - \_quickSpin, [80](#)
- CounterEventActivation
  - \_quickSpin, [80](#)

- CounterEventActivation\_AnyEdge  
CameraDefsC.h, [235](#)
- CounterEventActivation\_FallingEdge  
CameraDefsC.h, [235](#)
- CounterEventActivation\_LevelHigh  
CameraDefsC.h, [235](#)
- CounterEventActivation\_LevelLow  
CameraDefsC.h, [235](#)
- CounterEventActivation\_RisingEdge  
CameraDefsC.h, [235](#)
- CounterEventSource  
\_quickSpin, [81](#)
- CounterEventSource\_Counter0End  
CameraDefsC.h, [236](#)
- CounterEventSource\_Counter0Start  
CameraDefsC.h, [236](#)
- CounterEventSource\_Counter1End  
CameraDefsC.h, [236](#)
- CounterEventSource\_Counter1Start  
CameraDefsC.h, [236](#)
- CounterEventSource\_ExposureEnd  
CameraDefsC.h, [236](#)
- CounterEventSource\_ExposureStart  
CameraDefsC.h, [236](#)
- CounterEventSource\_FrameTriggerWait  
CameraDefsC.h, [236](#)
- CounterEventSource\_Line0  
CameraDefsC.h, [235](#)
- CounterEventSource\_Line1  
CameraDefsC.h, [236](#)
- CounterEventSource\_Line2  
CameraDefsC.h, [236](#)
- CounterEventSource\_Line3  
CameraDefsC.h, [236](#)
- CounterEventSource\_LogicBlock0  
CameraDefsC.h, [236](#)
- CounterEventSource\_LogicBlock1  
CameraDefsC.h, [236](#)
- CounterEventSource\_MHzTick  
CameraDefsC.h, [235](#)
- CounterEventSource\_Off  
CameraDefsC.h, [235](#)
- CounterEventSource\_UserOutput0  
CameraDefsC.h, [236](#)
- CounterEventSource\_UserOutput1  
CameraDefsC.h, [236](#)
- CounterEventSource\_UserOutput2  
CameraDefsC.h, [236](#)
- CounterEventSource\_UserOutput3  
CameraDefsC.h, [236](#)
- CounterReset  
\_quickSpin, [81](#)
- CounterResetActivation  
\_quickSpin, [81](#)
- CounterResetActivation\_AnyEdge  
CameraDefsC.h, [236](#)
- CounterResetActivation\_FallingEdge  
CameraDefsC.h, [236](#)
- CounterResetActivation\_LevelHigh  
CameraDefsC.h, [236](#)
- CounterResetActivation\_LevelLow  
CameraDefsC.h, [236](#)
- CounterResetActivation\_RisingEdge  
CameraDefsC.h, [236](#)
- CounterResetSource  
\_quickSpin, [81](#)
- CounterResetSource\_Counter0End  
CameraDefsC.h, [237](#)
- CounterResetSource\_Counter0Start  
CameraDefsC.h, [237](#)
- CounterResetSource\_Counter1End  
CameraDefsC.h, [237](#)
- CounterResetSource\_Counter1Start  
CameraDefsC.h, [237](#)
- CounterResetSource\_ExposureEnd  
CameraDefsC.h, [237](#)
- CounterResetSource\_ExposureStart  
CameraDefsC.h, [237](#)
- CounterResetSource\_FrameTriggerWait  
CameraDefsC.h, [237](#)
- CounterResetSource\_Line0  
CameraDefsC.h, [237](#)
- CounterResetSource\_Line1  
CameraDefsC.h, [237](#)
- CounterResetSource\_Line2  
CameraDefsC.h, [237](#)
- CounterResetSource\_Line3  
CameraDefsC.h, [237](#)
- CounterResetSource\_LogicBlock0  
CameraDefsC.h, [237](#)
- CounterResetSource\_LogicBlock1  
CameraDefsC.h, [237](#)
- CounterResetSource\_Off  
CameraDefsC.h, [236](#)
- CounterResetSource\_UserOutput0  
CameraDefsC.h, [237](#)
- CounterResetSource\_UserOutput1  
CameraDefsC.h, [237](#)
- CounterResetSource\_UserOutput2  
CameraDefsC.h, [237](#)
- CounterResetSource\_UserOutput3  
CameraDefsC.h, [237](#)
- CounterSelector  
\_quickSpin, [81](#)
- CounterSelector\_Counter0  
CameraDefsC.h, [237](#)
- CounterSelector\_Counter1  
CameraDefsC.h, [237](#)
- CounterStatus  
\_quickSpin, [81](#)
- CounterStatus\_CounterActive  
CameraDefsC.h, [237](#)
- CounterStatus\_CounterCompleted  
CameraDefsC.h, [237](#)
- CounterStatus\_CounterIdle  
CameraDefsC.h, [237](#)

- CounterStatus\_CounterOverflow
  - CameraDefsC.h, [237](#)
- CounterStatus\_CounterTriggerWait
  - CameraDefsC.h, [237](#)
- CounterTriggerActivation
  - \_quickSpin, [81](#)
- CounterTriggerActivation\_AnyEdge
  - CameraDefsC.h, [238](#)
- CounterTriggerActivation\_FallingEdge
  - CameraDefsC.h, [238](#)
- CounterTriggerActivation\_LevelHigh
  - CameraDefsC.h, [238](#)
- CounterTriggerActivation\_LevelLow
  - CameraDefsC.h, [238](#)
- CounterTriggerActivation\_RisingEdge
  - CameraDefsC.h, [238](#)
- CounterTriggerSource
  - \_quickSpin, [81](#)
- CounterTriggerSource\_Counter0End
  - CameraDefsC.h, [238](#)
- CounterTriggerSource\_Counter0Start
  - CameraDefsC.h, [238](#)
- CounterTriggerSource\_Counter1End
  - CameraDefsC.h, [238](#)
- CounterTriggerSource\_Counter1Start
  - CameraDefsC.h, [238](#)
- CounterTriggerSource\_ExposureEnd
  - CameraDefsC.h, [238](#)
- CounterTriggerSource\_ExposureStart
  - CameraDefsC.h, [238](#)
- CounterTriggerSource\_FrameTriggerWait
  - CameraDefsC.h, [238](#)
- CounterTriggerSource\_Line0
  - CameraDefsC.h, [238](#)
- CounterTriggerSource\_Line1
  - CameraDefsC.h, [238](#)
- CounterTriggerSource\_Line2
  - CameraDefsC.h, [238](#)
- CounterTriggerSource\_Line3
  - CameraDefsC.h, [238](#)
- CounterTriggerSource\_LogicBlock0
  - CameraDefsC.h, [238](#)
- CounterTriggerSource\_LogicBlock1
  - CameraDefsC.h, [238](#)
- CounterTriggerSource\_Off
  - CameraDefsC.h, [238](#)
- CounterTriggerSource\_UserOutput0
  - CameraDefsC.h, [238](#)
- CounterTriggerSource\_UserOutput1
  - CameraDefsC.h, [238](#)
- CounterTriggerSource\_UserOutput2
  - CameraDefsC.h, [238](#)
- CounterTriggerSource\_UserOutput3
  - CameraDefsC.h, [238](#)
- CounterValue
  - \_quickSpin, [82](#)
- CounterValueAtReset
  - \_quickSpin, [82](#)
- ctAllDependingNodes
  - SpinnakerGenApiDefsC.h, [460](#)
- ctAllTerminalNodes
  - SpinnakerGenApiDefsC.h, [460](#)
- ctDependingChildren
  - SpinnakerGenApiDefsC.h, [460](#)
- ctInvalidators
  - SpinnakerGenApiDefsC.h, [460](#)
- ctReadingChildren
  - SpinnakerGenApiDefsC.h, [460](#)
- ctWritingChildren
  - SpinnakerGenApiDefsC.h, [460](#)
- Custom
  - SpinnakerGenApiDefsC.h, [460](#)
- CxpConnectionSelector
  - \_quickSpin, [82](#)
- CxpConnectionTestErrorCount
  - \_quickSpin, [82](#)
- CxpConnectionTestMode
  - \_quickSpin, [82](#)
- CxpConnectionTestMode\_Mode1
  - CameraDefsC.h, [239](#)
- CxpConnectionTestMode\_Off
  - CameraDefsC.h, [239](#)
- CxpConnectionTestPacketCount
  - \_quickSpin, [82](#)
- CxpLinkConfiguration
  - \_quickSpin, [82](#)
- CxpLinkConfiguration\_Auto
  - CameraDefsC.h, [239](#)
- CxpLinkConfiguration\_CXP1\_X1
  - CameraDefsC.h, [239](#)
- CxpLinkConfiguration\_CXP1\_X2
  - CameraDefsC.h, [239](#)
- CxpLinkConfiguration\_CXP1\_X3
  - CameraDefsC.h, [239](#)
- CxpLinkConfiguration\_CXP1\_X4
  - CameraDefsC.h, [239](#)
- CxpLinkConfiguration\_CXP1\_X5
  - CameraDefsC.h, [240](#)
- CxpLinkConfiguration\_CXP1\_X6
  - CameraDefsC.h, [240](#)
- CxpLinkConfiguration\_CXP2\_X1
  - CameraDefsC.h, [239](#)
- CxpLinkConfiguration\_CXP2\_X2
  - CameraDefsC.h, [239](#)
- CxpLinkConfiguration\_CXP2\_X3
  - CameraDefsC.h, [239](#)
- CxpLinkConfiguration\_CXP2\_X4
  - CameraDefsC.h, [239](#)
- CxpLinkConfiguration\_CXP2\_X5
  - CameraDefsC.h, [240](#)
- CxpLinkConfiguration\_CXP2\_X6
  - CameraDefsC.h, [240](#)
- CxpLinkConfiguration\_CXP3\_X1
  - CameraDefsC.h, [239](#)
- CxpLinkConfiguration\_CXP3\_X2
  - CameraDefsC.h, [239](#)

- CxpLinkConfiguration\_CXP3\_X3  
CameraDefsC.h, [239](#)
- CxpLinkConfiguration\_CXP3\_X4  
CameraDefsC.h, [239](#)
- CxpLinkConfiguration\_CXP3\_X5  
CameraDefsC.h, [240](#)
- CxpLinkConfiguration\_CXP3\_X6  
CameraDefsC.h, [240](#)
- CxpLinkConfiguration\_CXP5\_X1  
CameraDefsC.h, [239](#)
- CxpLinkConfiguration\_CXP5\_X2  
CameraDefsC.h, [239](#)
- CxpLinkConfiguration\_CXP5\_X3  
CameraDefsC.h, [239](#)
- CxpLinkConfiguration\_CXP5\_X4  
CameraDefsC.h, [239](#)
- CxpLinkConfiguration\_CXP5\_X5  
CameraDefsC.h, [240](#)
- CxpLinkConfiguration\_CXP5\_X6  
CameraDefsC.h, [240](#)
- CxpLinkConfiguration\_CXP6\_X1  
CameraDefsC.h, [239](#)
- CxpLinkConfiguration\_CXP6\_X2  
CameraDefsC.h, [239](#)
- CxpLinkConfiguration\_CXP6\_X3  
CameraDefsC.h, [239](#)
- CxpLinkConfiguration\_CXP6\_X4  
CameraDefsC.h, [239](#)
- CxpLinkConfiguration\_CXP6\_X5  
CameraDefsC.h, [240](#)
- CxpLinkConfiguration\_CXP6\_X6  
CameraDefsC.h, [240](#)
- CxpLinkConfigurationPreferred  
\_quickSpin, [82](#)
- CxpLinkConfigurationPreferred\_CXP1\_X1  
CameraDefsC.h, [240](#)
- CxpLinkConfigurationPreferred\_CXP1\_X2  
CameraDefsC.h, [240](#)
- CxpLinkConfigurationPreferred\_CXP1\_X3  
CameraDefsC.h, [240](#)
- CxpLinkConfigurationPreferred\_CXP1\_X4  
CameraDefsC.h, [240](#)
- CxpLinkConfigurationPreferred\_CXP1\_X5  
CameraDefsC.h, [240](#)
- CxpLinkConfigurationPreferred\_CXP1\_X6  
CameraDefsC.h, [241](#)
- CxpLinkConfigurationPreferred\_CXP2\_X1  
CameraDefsC.h, [240](#)
- CxpLinkConfigurationPreferred\_CXP2\_X2  
CameraDefsC.h, [240](#)
- CxpLinkConfigurationPreferred\_CXP2\_X3  
CameraDefsC.h, [240](#)
- CxpLinkConfigurationPreferred\_CXP2\_X4  
CameraDefsC.h, [240](#)
- CxpLinkConfigurationPreferred\_CXP2\_X5  
CameraDefsC.h, [240](#)
- CxpLinkConfigurationPreferred\_CXP2\_X6  
CameraDefsC.h, [241](#)
- CxpLinkConfigurationPreferred\_CXP3\_X1  
CameraDefsC.h, [240](#)
- CxpLinkConfigurationPreferred\_CXP3\_X2  
CameraDefsC.h, [240](#)
- CxpLinkConfigurationPreferred\_CXP3\_X3  
CameraDefsC.h, [240](#)
- CxpLinkConfigurationPreferred\_CXP3\_X4  
CameraDefsC.h, [240](#)
- CxpLinkConfigurationPreferred\_CXP3\_X5  
CameraDefsC.h, [240](#)
- CxpLinkConfigurationPreferred\_CXP3\_X6  
CameraDefsC.h, [241](#)
- CxpLinkConfigurationPreferred\_CXP5\_X1  
CameraDefsC.h, [240](#)
- CxpLinkConfigurationPreferred\_CXP5\_X2  
CameraDefsC.h, [240](#)
- CxpLinkConfigurationPreferred\_CXP5\_X3  
CameraDefsC.h, [240](#)
- CxpLinkConfigurationPreferred\_CXP5\_X4  
CameraDefsC.h, [240](#)
- CxpLinkConfigurationPreferred\_CXP5\_X5  
CameraDefsC.h, [240](#)
- CxpLinkConfigurationPreferred\_CXP5\_X6  
CameraDefsC.h, [241](#)
- CxpLinkConfigurationPreferred\_CXP6\_X1  
CameraDefsC.h, [240](#)
- CxpLinkConfigurationPreferred\_CXP6\_X2  
CameraDefsC.h, [240](#)
- CxpLinkConfigurationPreferred\_CXP6\_X3  
CameraDefsC.h, [240](#)
- CxpLinkConfigurationPreferred\_CXP6\_X4  
CameraDefsC.h, [240](#)
- CxpLinkConfigurationPreferred\_CXP6\_X5  
CameraDefsC.h, [241](#)
- CxpLinkConfigurationPreferred\_CXP6\_X6  
CameraDefsC.h, [241](#)
- CxpLinkConfigurationStatus  
\_quickSpin, [83](#)
- CxpLinkConfigurationStatus\_CXP1\_X1  
CameraDefsC.h, [241](#)
- CxpLinkConfigurationStatus\_CXP1\_X2  
CameraDefsC.h, [241](#)
- CxpLinkConfigurationStatus\_CXP1\_X3  
CameraDefsC.h, [241](#)
- CxpLinkConfigurationStatus\_CXP1\_X4  
CameraDefsC.h, [241](#)
- CxpLinkConfigurationStatus\_CXP1\_X5  
CameraDefsC.h, [241](#)
- CxpLinkConfigurationStatus\_CXP1\_X6  
CameraDefsC.h, [242](#)
- CxpLinkConfigurationStatus\_CXP2\_X1  
CameraDefsC.h, [241](#)
- CxpLinkConfigurationStatus\_CXP2\_X2  
CameraDefsC.h, [241](#)
- CxpLinkConfigurationStatus\_CXP2\_X3  
CameraDefsC.h, [241](#)
- CxpLinkConfigurationStatus\_CXP2\_X4  
CameraDefsC.h, [241](#)

- CxpLinkConfigurationStatus\_CXP2\_X5
  - CameraDefsC.h, [241](#)
- CxpLinkConfigurationStatus\_CXP2\_X6
  - CameraDefsC.h, [242](#)
- CxpLinkConfigurationStatus\_CXP3\_X1
  - CameraDefsC.h, [241](#)
- CxpLinkConfigurationStatus\_CXP3\_X2
  - CameraDefsC.h, [241](#)
- CxpLinkConfigurationStatus\_CXP3\_X3
  - CameraDefsC.h, [241](#)
- CxpLinkConfigurationStatus\_CXP3\_X4
  - CameraDefsC.h, [241](#)
- CxpLinkConfigurationStatus\_CXP3\_X5
  - CameraDefsC.h, [241](#)
- CxpLinkConfigurationStatus\_CXP3\_X6
  - CameraDefsC.h, [242](#)
- CxpLinkConfigurationStatus\_CXP5\_X1
  - CameraDefsC.h, [241](#)
- CxpLinkConfigurationStatus\_CXP5\_X2
  - CameraDefsC.h, [241](#)
- CxpLinkConfigurationStatus\_CXP5\_X3
  - CameraDefsC.h, [241](#)
- CxpLinkConfigurationStatus\_CXP5\_X4
  - CameraDefsC.h, [241](#)
- CxpLinkConfigurationStatus\_CXP5\_X5
  - CameraDefsC.h, [241](#)
- CxpLinkConfigurationStatus\_CXP5\_X6
  - CameraDefsC.h, [242](#)
- CxpLinkConfigurationStatus\_CXP6\_X1
  - CameraDefsC.h, [241](#)
- CxpLinkConfigurationStatus\_CXP6\_X2
  - CameraDefsC.h, [241](#)
- CxpLinkConfigurationStatus\_CXP6\_X3
  - CameraDefsC.h, [241](#)
- CxpLinkConfigurationStatus\_CXP6\_X4
  - CameraDefsC.h, [241](#)
- CxpLinkConfigurationStatus\_CXP6\_X5
  - CameraDefsC.h, [242](#)
- CxpLinkConfigurationStatus\_CXP6\_X6
  - CameraDefsC.h, [242](#)
- CxpLinkConfigurationStatus\_None
  - CameraDefsC.h, [241](#)
- CxpLinkConfigurationStatus\_Pending
  - CameraDefsC.h, [241](#)
- CxpPoCxpAuto
  - \_quickSpin, [83](#)
- CxpPoCxpStatus
  - \_quickSpin, [83](#)
- CxpPoCxpStatus\_Auto
  - CameraDefsC.h, [242](#)
- CxpPoCxpStatus\_Off
  - CameraDefsC.h, [242](#)
- CxpPoCxpStatus\_Tripped
  - CameraDefsC.h, [242](#)
- CxpPoCxpTripReset
  - \_quickSpin, [83](#)
- CxpPoCxpTurnOff
  - \_quickSpin, [83](#)
- DecimationHorizontal
  - \_quickSpin, [83](#)
- DecimationHorizontalMode
  - \_quickSpin, [83](#)
- DecimationHorizontalMode\_Discard
  - CameraDefsC.h, [242](#)
- DecimationSelector
  - \_quickSpin, [83](#)
- DecimationSelector\_All
  - CameraDefsC.h, [243](#)
- DecimationSelector\_Sensor
  - CameraDefsC.h, [243](#)
- DecimationVertical
  - \_quickSpin, [84](#)
- DecimationVerticalMode
  - \_quickSpin, [84](#)
- DecimationVerticalMode\_Discard
  - CameraDefsC.h, [243](#)
- Decreasing
  - SpinnakerGenApiDefsC.h, [462](#)
- DEFAULT
  - SpinnakerDefsC.h, [409](#)
- DefectCorrectionMode
  - \_quickSpin, [84](#)
- DefectCorrectionMode\_Average
  - CameraDefsC.h, [243](#)
- DefectCorrectionMode\_Highlight
  - CameraDefsC.h, [243](#)
- DefectCorrectionMode\_Zero
  - CameraDefsC.h, [243](#)
- DefectCorrectStaticEnable
  - \_quickSpin, [84](#)
- DefectTableApply
  - \_quickSpin, [84](#)
- DefectTableCoordinateX
  - \_quickSpin, [84](#)
- DefectTableCoordinateY
  - \_quickSpin, [84](#)
- DefectTableFactoryRestore
  - \_quickSpin, [84](#)
- DefectTableIndex
  - \_quickSpin, [85](#)
- DefectTablePixelCount
  - \_quickSpin, [85](#)
- DefectTableSave
  - \_quickSpin, [85](#)
- DEFLATE
  - SpinnakerDefsC.h, [413](#)
- Deinterlacing
  - \_quickSpin, [85](#)
- Deinterlacing\_LineDuplication
  - CameraDefsC.h, [244](#)
- Deinterlacing\_Off
  - CameraDefsC.h, [244](#)
- Deinterlacing\_Weave
  - CameraDefsC.h, [244](#)
- Device Event Data Access, [23](#)
- DeviceAccessStatus

- [\\_quickSpinTLDevice](#), 149
- [\\_quickSpinTLInterface](#), 155
- [DeviceAccessStatus\\_Busy](#)
  - [TransportLayerDefsC.h](#), 469
- [DeviceAccessStatus\\_NoAccess](#)
  - [TransportLayerDefsC.h](#), 469
- [DeviceAccessStatus\\_OpenReadOnly](#)
  - [TransportLayerDefsC.h](#), 469
- [DeviceAccessStatus\\_OpenReadWrite](#)
  - [TransportLayerDefsC.h](#), 469
- [DeviceAccessStatus\\_ReadOnly](#)
  - [TransportLayerDefsC.h](#), 469
- [DeviceAccessStatus\\_ReadWrite](#)
  - [TransportLayerDefsC.h](#), 469
- [DeviceAccessStatus\\_Unknown](#)
  - [TransportLayerDefsC.h](#), 469
- [DeviceAddress](#)
  - [\\_actionCommandResult](#), 51
- [DeviceCharacterSet](#)
  - [\\_quickSpin](#), 85
- [DeviceCharacterSet\\_ASCII](#)
  - [CameraDefsC.h](#), 244
- [DeviceCharacterSet\\_UTF8](#)
  - [CameraDefsC.h](#), 244
- [DeviceClockFrequency](#)
  - [\\_quickSpin](#), 85
- [DeviceClockSelector](#)
  - [\\_quickSpin](#), 85
- [DeviceClockSelector\\_CameraLink](#)
  - [CameraDefsC.h](#), 244
- [DeviceClockSelector\\_Sensor](#)
  - [CameraDefsC.h](#), 244
- [DeviceClockSelector\\_SensorDigitization](#)
  - [CameraDefsC.h](#), 244
- [DeviceConnectionSelector](#)
  - [\\_quickSpin](#), 85
- [DeviceConnectionSpeed](#)
  - [\\_quickSpin](#), 86
- [DeviceConnectionStatus](#)
  - [\\_quickSpin](#), 86
- [DeviceConnectionStatus\\_Active](#)
  - [CameraDefsC.h](#), 244
- [DeviceConnectionStatus\\_Inactive](#)
  - [CameraDefsC.h](#), 244
- [DeviceCount](#)
  - [\\_quickSpinTLInterface](#), 155
- [DeviceCurrentSpeed](#)
  - [\\_quickSpinTLDevice](#), 149
- [DeviceCurrentSpeed\\_FullSpeed](#)
  - [TransportLayerDefsC.h](#), 469
- [DeviceCurrentSpeed\\_HighSpeed](#)
  - [TransportLayerDefsC.h](#), 469
- [DeviceCurrentSpeed\\_LowSpeed](#)
  - [TransportLayerDefsC.h](#), 469
- [DeviceCurrentSpeed\\_SuperSpeed](#)
  - [TransportLayerDefsC.h](#), 469
- [DeviceCurrentSpeed\\_UnknownSpeed](#)
  - [TransportLayerDefsC.h](#), 469
- [DeviceDisplayName](#)
  - [\\_quickSpinTLDevice](#), 149
- [DeviceDriverVersion](#)
  - [\\_quickSpinTLDevice](#), 149
- [DeviceEndianessMechanism](#)
  - [\\_quickSpinTLDevice](#), 149
- [DeviceEndianessMechanism\\_Legacy](#)
  - [TransportLayerDefsC.h](#), 471
- [DeviceEndianessMechanism\\_Standard](#)
  - [TransportLayerDefsC.h](#), 471
- [DeviceEventChannelCount](#)
  - [\\_quickSpin](#), 86
- [DeviceFamilyName](#)
  - [\\_quickSpin](#), 86
- [DeviceFeaturePersistenceEnd](#)
  - [\\_quickSpin](#), 86
- [DeviceFeaturePersistenceStart](#)
  - [\\_quickSpin](#), 86
- [DeviceFirmwareVersion](#)
  - [\\_quickSpin](#), 86
- [DeviceGenCPVersionMajor](#)
  - [\\_quickSpin](#), 86
- [DeviceGenCPVersionMinor](#)
  - [\\_quickSpin](#), 87
- [DeviceID](#)
  - [\\_quickSpin](#), 87
  - [\\_quickSpinTLDevice](#), 150
  - [\\_quickSpinTLInterface](#), 156
- [DeviceIndicatorMode](#)
  - [\\_quickSpin](#), 87
- [DeviceIndicatorMode\\_Active](#)
  - [CameraDefsC.h](#), 245
- [DeviceIndicatorMode\\_ErrorStatus](#)
  - [CameraDefsC.h](#), 245
- [DeviceIndicatorMode\\_Inactive](#)
  - [CameraDefsC.h](#), 245
- [DeviceInstanceId](#)
  - [\\_quickSpinTLDevice](#), 150
- [DeviceIsUpdater](#)
  - [\\_quickSpinTLDevice](#), 150
- [DeviceLinkBandwidthReserve](#)
  - [\\_quickSpin](#), 87
- [DeviceLinkCommandTimeout](#)
  - [\\_quickSpin](#), 87
- [DeviceLinkConnectionCount](#)
  - [\\_quickSpin](#), 87
- [DeviceLinkCurrentThroughput](#)
  - [\\_quickSpin](#), 87
- [DeviceLinkHeartbeatMode](#)
  - [\\_quickSpin](#), 87
- [DeviceLinkHeartbeatMode\\_Off](#)
  - [CameraDefsC.h](#), 245
- [DeviceLinkHeartbeatMode\\_On](#)
  - [CameraDefsC.h](#), 245
- [DeviceLinkHeartbeatTimeout](#)
  - [\\_quickSpin](#), 88
- [DeviceLinkSelector](#)
  - [\\_quickSpin](#), 88



- DeviceLinkSpeed
  - [\\_quickSpin](#), 88
  - [\\_quickSpinTLDevice](#), 150
- DeviceLinkThroughputLimit
  - [\\_quickSpin](#), 88
- DeviceLinkThroughputLimitMode
  - [\\_quickSpin](#), 88
- DeviceLinkThroughputLimitMode\_Off
  - CameraDefsC.h, 245
- DeviceLinkThroughputLimitMode\_On
  - CameraDefsC.h, 245
- DeviceLocation
  - [\\_quickSpinTLDevice](#), 150
- DeviceManifestEntrySelector
  - [\\_quickSpin](#), 88
- DeviceManifestPrimaryURL
  - [\\_quickSpin](#), 88
- DeviceManifestSchemaMajorVersion
  - [\\_quickSpin](#), 88
- DeviceManifestSchemaMinorVersion
  - [\\_quickSpin](#), 89
- DeviceManifestSecondaryURL
  - [\\_quickSpin](#), 89
- DeviceManifestXMLMajorVersion
  - [\\_quickSpin](#), 89
- DeviceManifestXMLMinorVersion
  - [\\_quickSpin](#), 89
- DeviceManifestXMLSubMinorVersion
  - [\\_quickSpin](#), 89
- DeviceManufacturerInfo
  - [\\_quickSpin](#), 89
- DeviceMaxThroughput
  - [\\_quickSpin](#), 89
- DeviceModelName
  - [\\_quickSpin](#), 89
  - [\\_quickSpinTLDevice](#), 150
  - [\\_quickSpinTLInterface](#), 156
- DeviceMulticastMonitorMode
  - [\\_quickSpinTLDevice](#), 150
- DevicePowerSupplySelector
  - [\\_quickSpin](#), 90
- DevicePowerSupplySelector\_External
  - CameraDefsC.h, 246
- DeviceRegistersCheck
  - [\\_quickSpin](#), 90
- DeviceRegistersEndianness
  - [\\_quickSpin](#), 90
- DeviceRegistersEndianness\_Big
  - CameraDefsC.h, 246
- DeviceRegistersEndianness\_Little
  - CameraDefsC.h, 246
- DeviceRegistersStreamingEnd
  - [\\_quickSpin](#), 90
- DeviceRegistersStreamingStart
  - [\\_quickSpin](#), 90
- DeviceRegistersValid
  - [\\_quickSpin](#), 90
- DeviceReset
  - [\\_quickSpin](#), 90
- DeviceScanType
  - [\\_quickSpin](#), 90
- DeviceScanType\_Areascan
  - CameraDefsC.h, 246
- DeviceSelector
  - [\\_quickSpinTLInterface](#), 156
- DeviceSerialNumber
  - [\\_quickSpin](#), 91
  - [\\_quickSpinTLDevice](#), 150
  - [\\_quickSpinTLInterface](#), 156
- DeviceSerialPortBaudRate
  - [\\_quickSpin](#), 91
- DeviceSerialPortBaudRate\_Baud115200
  - CameraDefsC.h, 247
- DeviceSerialPortBaudRate\_Baud19200
  - CameraDefsC.h, 247
- DeviceSerialPortBaudRate\_Baud230400
  - CameraDefsC.h, 247
- DeviceSerialPortBaudRate\_Baud38400
  - CameraDefsC.h, 247
- DeviceSerialPortBaudRate\_Baud460800
  - CameraDefsC.h, 247
- DeviceSerialPortBaudRate\_Baud57600
  - CameraDefsC.h, 247
- DeviceSerialPortBaudRate\_Baud921600
  - CameraDefsC.h, 247
- DeviceSerialPortBaudRate\_Baud9600
  - CameraDefsC.h, 247
- DeviceSerialPortSelector
  - [\\_quickSpin](#), 91
- DeviceSerialPortSelector\_CameraLink
  - CameraDefsC.h, 247
- DeviceSFNCVersionMajor
  - [\\_quickSpin](#), 91
- DeviceSFNCVersionMinor
  - [\\_quickSpin](#), 91
- DeviceSFNCVersionSubMinor
  - [\\_quickSpin](#), 91
- DeviceStreamChannelCount
  - [\\_quickSpin](#), 91
- DeviceStreamChannelEndianness
  - [\\_quickSpin](#), 91
- DeviceStreamChannelEndianness\_Big
  - CameraDefsC.h, 247
- DeviceStreamChannelEndianness\_Little
  - CameraDefsC.h, 247
- DeviceStreamChannelLink
  - [\\_quickSpin](#), 92
- DeviceStreamChannelPacketSize
  - [\\_quickSpin](#), 92
- DeviceStreamChannelSelector
  - [\\_quickSpin](#), 92
- DeviceStreamChannelType
  - [\\_quickSpin](#), 92
- DeviceStreamChannelType\_Receiver
  - CameraDefsC.h, 248
- DeviceStreamChannelType\_Transmitter

- CameraDefsC.h, [248](#)
- DeviceTapGeometry
  - \_quickSpin, [92](#)
- DeviceTapGeometry\_Geometry\_10X
  - CameraDefsC.h, [249](#)
- DeviceTapGeometry\_Geometry\_10X\_1Y
  - CameraDefsC.h, [249](#)
- DeviceTapGeometry\_Geometry\_1X
  - CameraDefsC.h, [248](#)
- DeviceTapGeometry\_Geometry\_1X10
  - CameraDefsC.h, [249](#)
- DeviceTapGeometry\_Geometry\_1X10\_1Y
  - CameraDefsC.h, [249](#)
- DeviceTapGeometry\_Geometry\_1X2
  - CameraDefsC.h, [248](#)
- DeviceTapGeometry\_Geometry\_1X2\_1Y
  - CameraDefsC.h, [248](#)
- DeviceTapGeometry\_Geometry\_1X2\_1Y2
  - CameraDefsC.h, [248](#)
- DeviceTapGeometry\_Geometry\_1X2\_2YE
  - CameraDefsC.h, [248](#)
- DeviceTapGeometry\_Geometry\_1X3
  - CameraDefsC.h, [248](#)
- DeviceTapGeometry\_Geometry\_1X3\_1Y
  - CameraDefsC.h, [248](#)
- DeviceTapGeometry\_Geometry\_1X4
  - CameraDefsC.h, [248](#)
- DeviceTapGeometry\_Geometry\_1X4\_1Y
  - CameraDefsC.h, [248](#)
- DeviceTapGeometry\_Geometry\_1X8
  - CameraDefsC.h, [249](#)
- DeviceTapGeometry\_Geometry\_1X8\_1Y
  - CameraDefsC.h, [249](#)
- DeviceTapGeometry\_Geometry\_1X\_1Y
  - CameraDefsC.h, [248](#)
- DeviceTapGeometry\_Geometry\_1X\_1Y2
  - CameraDefsC.h, [248](#)
- DeviceTapGeometry\_Geometry\_1X\_2YE
  - CameraDefsC.h, [248](#)
- DeviceTapGeometry\_Geometry\_2X
  - CameraDefsC.h, [248](#)
- DeviceTapGeometry\_Geometry\_2X2
  - CameraDefsC.h, [248](#)
- DeviceTapGeometry\_Geometry\_2X2\_1Y
  - CameraDefsC.h, [248](#)
- DeviceTapGeometry\_Geometry\_2X2E
  - CameraDefsC.h, [248](#)
- DeviceTapGeometry\_Geometry\_2X2E\_1Y
  - CameraDefsC.h, [248](#)
- DeviceTapGeometry\_Geometry\_2X2E\_2YE
  - CameraDefsC.h, [249](#)
- DeviceTapGeometry\_Geometry\_2X2M
  - CameraDefsC.h, [249](#)
- DeviceTapGeometry\_Geometry\_2X\_1Y
  - CameraDefsC.h, [248](#)
- DeviceTapGeometry\_Geometry\_2X\_1Y2
  - CameraDefsC.h, [248](#)
- DeviceTapGeometry\_Geometry\_2X\_2YE
  - CameraDefsC.h, [248](#)
- DeviceTapGeometry\_Geometry\_2XE
  - CameraDefsC.h, [248](#)
- DeviceTapGeometry\_Geometry\_2XE\_1Y
  - CameraDefsC.h, [248](#)
- DeviceTapGeometry\_Geometry\_2XE\_2YE
  - CameraDefsC.h, [248](#)
- DeviceTapGeometry\_Geometry\_2XM
  - CameraDefsC.h, [248](#)
- DeviceTapGeometry\_Geometry\_2XM\_1Y
  - CameraDefsC.h, [248](#)
- DeviceTapGeometry\_Geometry\_2XM\_1Y2
  - CameraDefsC.h, [248](#)
- DeviceTapGeometry\_Geometry\_2XM\_2YE
  - CameraDefsC.h, [248](#)
- DeviceTapGeometry\_Geometry\_3X
  - CameraDefsC.h, [248](#)
- DeviceTapGeometry\_Geometry\_3X\_1Y
  - CameraDefsC.h, [248](#)
- DeviceTapGeometry\_Geometry\_4X
  - CameraDefsC.h, [248](#)
- DeviceTapGeometry\_Geometry\_4X2
  - CameraDefsC.h, [249](#)
- DeviceTapGeometry\_Geometry\_4X2\_1Y
  - CameraDefsC.h, [249](#)
- DeviceTapGeometry\_Geometry\_4X2E
  - CameraDefsC.h, [249](#)
- DeviceTapGeometry\_Geometry\_4X2E\_1Y
  - CameraDefsC.h, [249](#)
- DeviceTapGeometry\_Geometry\_4X\_1Y
  - CameraDefsC.h, [248](#)
- DeviceTapGeometry\_Geometry\_8X
  - CameraDefsC.h, [249](#)
- DeviceTapGeometry\_Geometry\_8X\_1Y
  - CameraDefsC.h, [249](#)
- DeviceTemperature
  - \_quickSpin, [92](#)
- DeviceTemperatureSelector
  - \_quickSpin, [92](#)
- DeviceTemperatureSelector\_Sensor
  - CameraDefsC.h, [249](#)
- DeviceTLType
  - \_quickSpin, [92](#)
- DeviceTLType\_CameraLink
  - CameraDefsC.h, [249](#)
- DeviceTLType\_CameraLinkHS
  - CameraDefsC.h, [249](#)
- DeviceTLType\_CoaXPRESS
  - CameraDefsC.h, [249](#)
- DeviceTLType\_Custom
  - CameraDefsC.h, [249](#)
- DeviceTLType\_GigEVision
  - CameraDefsC.h, [249](#)
- DeviceTLType\_USB3Vision
  - CameraDefsC.h, [249](#)
- DeviceTLVersionMajor
  - \_quickSpin, [93](#)
- DeviceTLVersionMinor



- [\\_quickSpin](#), 93
- DeviceTLVersionSubMinor
  - [\\_quickSpin](#), 93
- DeviceType
  - [\\_quickSpin](#), 93
  - [\\_quickSpinTLDevice](#), 151
- DeviceType\_CameraLink
  - TransportLayerDefsC.h, 471
- DeviceType\_CameraLinkHS
  - TransportLayerDefsC.h, 471
- DeviceType\_CoaXPress
  - TransportLayerDefsC.h, 471
- DeviceType\_Custom
  - TransportLayerDefsC.h, 471
- DeviceType\_GigEVision
  - TransportLayerDefsC.h, 471
- DeviceType\_Peripheral
  - CameraDefsC.h, 250
- DeviceType\_Receiver
  - CameraDefsC.h, 250
- DeviceType\_Transceiver
  - CameraDefsC.h, 250
- DeviceType\_Transmitter
  - CameraDefsC.h, 250
- DeviceType\_USB3Vision
  - TransportLayerDefsC.h, 471
- DeviceU3VProtocol
  - [\\_quickSpinTLDevice](#), 151
- DeviceUnlock
  - [\\_quickSpinTLInterface](#), 156
- DeviceUpdateList
  - [\\_quickSpinTLInterface](#), 156
- DeviceUptime
  - [\\_quickSpin](#), 93
- DeviceUserID
  - [\\_quickSpin](#), 93
  - [\\_quickSpinTLDevice](#), 151
- DeviceVendorName
  - [\\_quickSpin](#), 93
  - [\\_quickSpinTLDevice](#), 151
  - [\\_quickSpinTLInterface](#), 156
- DeviceVersion
  - [\\_quickSpin](#), 93
  - [\\_quickSpinTLDevice](#), 151
- DIRECTIONAL\_FILTER
  - SpinnakerDefsC.h, 409
- EDGE\_SENSING
  - SpinnakerDefsC.h, 409
- EncoderDivider
  - [\\_quickSpin](#), 94
- EncoderMode
  - [\\_quickSpin](#), 94
- EncoderMode\_FourPhase
  - CameraDefsC.h, 250
- EncoderMode\_HighResolution
  - CameraDefsC.h, 250
- EncoderOutputMode
  - [\\_quickSpin](#), 94
- EncoderOutputMode\_DirectionDown
  - CameraDefsC.h, 251
- EncoderOutputMode\_DirectionUp
  - CameraDefsC.h, 251
- EncoderOutputMode\_Motion
  - CameraDefsC.h, 251
- EncoderOutputMode\_Off
  - CameraDefsC.h, 250
- EncoderOutputMode\_PositionDown
  - CameraDefsC.h, 251
- EncoderOutputMode\_PositionUp
  - CameraDefsC.h, 250
- EncoderReset
  - [\\_quickSpin](#), 94
- EncoderResetActivation
  - [\\_quickSpin](#), 94
- EncoderResetActivation\_AnyEdge
  - CameraDefsC.h, 251
- EncoderResetActivation\_FallingEdge
  - CameraDefsC.h, 251
- EncoderResetActivation\_LevelHigh
  - CameraDefsC.h, 251
- EncoderResetActivation\_LevelLow
  - CameraDefsC.h, 251
- EncoderResetActivation\_RisingEdge
  - CameraDefsC.h, 251
- EncoderResetSource
  - [\\_quickSpin](#), 94
- EncoderResetSource\_AcquisitionEnd
  - CameraDefsC.h, 251
- EncoderResetSource\_AcquisitionStart
  - CameraDefsC.h, 251
- EncoderResetSource\_AcquisitionTrigger
  - CameraDefsC.h, 251
- EncoderResetSource\_Action0
  - CameraDefsC.h, 252
- EncoderResetSource\_Action1
  - CameraDefsC.h, 252
- EncoderResetSource\_Action2
  - CameraDefsC.h, 252
- EncoderResetSource\_Counter0End
  - CameraDefsC.h, 252
- EncoderResetSource\_Counter0Start
  - CameraDefsC.h, 252
- EncoderResetSource\_Counter1End
  - CameraDefsC.h, 252
- EncoderResetSource\_Counter1Start
  - CameraDefsC.h, 252
- EncoderResetSource\_Counter2End
  - CameraDefsC.h, 252
- EncoderResetSource\_Counter2Start
  - CameraDefsC.h, 252
- EncoderResetSource\_ExposureEnd
  - CameraDefsC.h, 252
- EncoderResetSource\_ExposureStart
  - CameraDefsC.h, 251
- EncoderResetSource\_FrameEnd
  - CameraDefsC.h, 251

EncoderResetSource\_FrameStart  
CameraDefsC.h, [251](#)

EncoderResetSource\_FrameTrigger  
CameraDefsC.h, [251](#)

EncoderResetSource\_Line0  
CameraDefsC.h, [252](#)

EncoderResetSource\_Line1  
CameraDefsC.h, [252](#)

EncoderResetSource\_Line2  
CameraDefsC.h, [252](#)

EncoderResetSource\_LinkTrigger0  
CameraDefsC.h, [252](#)

EncoderResetSource\_LinkTrigger1  
CameraDefsC.h, [252](#)

EncoderResetSource\_LinkTrigger2  
CameraDefsC.h, [252](#)

EncoderResetSource\_Off  
CameraDefsC.h, [251](#)

EncoderResetSource\_SoftwareSignal0  
CameraDefsC.h, [252](#)

EncoderResetSource\_SoftwareSignal1  
CameraDefsC.h, [252](#)

EncoderResetSource\_SoftwareSignal2  
CameraDefsC.h, [252](#)

EncoderResetSource\_Timer0End  
CameraDefsC.h, [252](#)

EncoderResetSource\_Timer0Start  
CameraDefsC.h, [252](#)

EncoderResetSource\_Timer1End  
CameraDefsC.h, [252](#)

EncoderResetSource\_Timer1Start  
CameraDefsC.h, [252](#)

EncoderResetSource\_Timer2End  
CameraDefsC.h, [252](#)

EncoderResetSource\_Timer2Start  
CameraDefsC.h, [252](#)

EncoderResetSource\_UserOutput0  
CameraDefsC.h, [252](#)

EncoderResetSource\_UserOutput1  
CameraDefsC.h, [252](#)

EncoderResetSource\_UserOutput2  
CameraDefsC.h, [252](#)

EncoderSelector  
\_quickSpin, [94](#)

EncoderSelector\_Encoder0  
CameraDefsC.h, [252](#)

EncoderSelector\_Encoder1  
CameraDefsC.h, [252](#)

EncoderSelector\_Encoder2  
CameraDefsC.h, [252](#)

EncoderSourceA  
\_quickSpin, [94](#)

EncoderSourceA\_Line0  
CameraDefsC.h, [253](#)

EncoderSourceA\_Line1  
CameraDefsC.h, [253](#)

EncoderSourceA\_Line2  
CameraDefsC.h, [253](#)

EncoderSourceA\_Off  
CameraDefsC.h, [253](#)

EncoderSourceB  
\_quickSpin, [95](#)

EncoderSourceB\_Line0  
CameraDefsC.h, [253](#)

EncoderSourceB\_Line1  
CameraDefsC.h, [253](#)

EncoderSourceB\_Line2  
CameraDefsC.h, [253](#)

EncoderSourceB\_Off  
CameraDefsC.h, [253](#)

EncoderStatus  
\_quickSpin, [95](#)

EncoderStatus\_EncoderDown  
CameraDefsC.h, [253](#)

EncoderStatus\_EncoderIdle  
CameraDefsC.h, [253](#)

EncoderStatus\_EncoderStatic  
CameraDefsC.h, [253](#)

EncoderStatus\_EncoderUp  
CameraDefsC.h, [253](#)

EncoderTimeout  
\_quickSpin, [95](#)

EncoderValue  
\_quickSpin, [95](#)

EncoderValueAtReset  
\_quickSpin, [95](#)

EnumEntryNode  
SpinnakerGenApiDefsC.h, [461](#)

EnumerateGEVInterfaces  
\_quickSpinTLSystem, [166](#)

EnumerationCount  
\_quickSpin, [95](#)

EnumerationNode  
SpinnakerGenApiDefsC.h, [461](#)

Error Handling, [13](#)

Event Access, [20](#)

EventAcquisitionEnd  
\_quickSpin, [95](#)

EventAcquisitionEndFrameID  
\_quickSpin, [95](#)

EventAcquisitionEndTimestamp  
\_quickSpin, [96](#)

EventAcquisitionError  
\_quickSpin, [96](#)

EventAcquisitionErrorFrameID  
\_quickSpin, [96](#)

EventAcquisitionErrorTimestamp  
\_quickSpin, [96](#)

EventAcquisitionStart  
\_quickSpin, [96](#)

EventAcquisitionStartFrameID  
\_quickSpin, [96](#)

EventAcquisitionStartTimestamp  
\_quickSpin, [96](#)

EventAcquisitionTransferEnd  
\_quickSpin, [96](#)

EventAcquisitionTransferEndFrameID  
    \_quickSpin, [97](#)  
EventAcquisitionTransferEndTimestamp  
    \_quickSpin, [97](#)  
EventAcquisitionTransferStart  
    \_quickSpin, [97](#)  
EventAcquisitionTransferStartFrameID  
    \_quickSpin, [97](#)  
EventAcquisitionTransferStartTimestamp  
    \_quickSpin, [97](#)  
EventAcquisitionTrigger  
    \_quickSpin, [97](#)  
EventAcquisitionTriggerFrameID  
    \_quickSpin, [97](#)  
EventAcquisitionTriggerTimestamp  
    \_quickSpin, [97](#)  
EventActionLate  
    \_quickSpin, [98](#)  
EventActionLateFrameID  
    \_quickSpin, [98](#)  
EventActionLateTimestamp  
    \_quickSpin, [98](#)  
EventCounter0End  
    \_quickSpin, [98](#)  
EventCounter0EndFrameID  
    \_quickSpin, [98](#)  
EventCounter0EndTimestamp  
    \_quickSpin, [98](#)  
EventCounter0Start  
    \_quickSpin, [98](#)  
EventCounter0StartFrameID  
    \_quickSpin, [98](#)  
EventCounter0StartTimestamp  
    \_quickSpin, [99](#)  
EventCounter1End  
    \_quickSpin, [99](#)  
EventCounter1EndFrameID  
    \_quickSpin, [99](#)  
EventCounter1EndTimestamp  
    \_quickSpin, [99](#)  
EventCounter1Start  
    \_quickSpin, [99](#)  
EventCounter1StartFrameID  
    \_quickSpin, [99](#)  
EventCounter1StartTimestamp  
    \_quickSpin, [99](#)  
EventEncoder0Restarted  
    \_quickSpin, [99](#)  
EventEncoder0RestartedFrameID  
    \_quickSpin, [100](#)  
EventEncoder0RestartedTimestamp  
    \_quickSpin, [100](#)  
EventEncoder0Stopped  
    \_quickSpin, [100](#)  
EventEncoder0StoppedFrameID  
    \_quickSpin, [100](#)  
EventEncoder0StoppedTimestamp  
    \_quickSpin, [100](#)  
EventEncoder1Restarted  
    \_quickSpin, [100](#)  
EventEncoder1RestartedFrameID  
    \_quickSpin, [100](#)  
EventEncoder1RestartedTimestamp  
    \_quickSpin, [100](#)  
EventEncoder1Stopped  
    \_quickSpin, [101](#)  
EventEncoder1StoppedFrameID  
    \_quickSpin, [101](#)  
EventEncoder1StoppedTimestamp  
    \_quickSpin, [101](#)  
EventError  
    \_quickSpin, [101](#)  
EventErrorCode  
    \_quickSpin, [101](#)  
EventErrorFrameID  
    \_quickSpin, [101](#)  
EventErrorTimestamp  
    \_quickSpin, [101](#)  
EventExposureEnd  
    \_quickSpin, [101](#)  
EventExposureEndFrameID  
    \_quickSpin, [102](#)  
EventExposureEndTimestamp  
    \_quickSpin, [102](#)  
EventExposureStart  
    \_quickSpin, [102](#)  
EventExposureStartFrameID  
    \_quickSpin, [102](#)  
EventExposureStartTimestamp  
    \_quickSpin, [102](#)  
EventFrameBurstEnd  
    \_quickSpin, [102](#)  
EventFrameBurstEndFrameID  
    \_quickSpin, [102](#)  
EventFrameBurstEndTimestamp  
    \_quickSpin, [102](#)  
EventFrameBurstStart  
    \_quickSpin, [103](#)  
EventFrameBurstStartFrameID  
    \_quickSpin, [103](#)  
EventFrameBurstStartTimestamp  
    \_quickSpin, [103](#)  
EventFrameEnd  
    \_quickSpin, [103](#)  
EventFrameEndFrameID  
    \_quickSpin, [103](#)  
EventFrameEndTimestamp  
    \_quickSpin, [103](#)  
EventFrameStart  
    \_quickSpin, [103](#)  
EventFrameStartFrameID  
    \_quickSpin, [103](#)  
EventFrameStartTimestamp  
    \_quickSpin, [104](#)  
EventFrameTransferEnd  
    \_quickSpin, [104](#)

EventFrameTransferEndFrameID  
     \_quickSpin, [104](#)  
 EventFrameTransferEndTimestamp  
     \_quickSpin, [104](#)  
 EventFrameTransferStart  
     \_quickSpin, [104](#)  
 EventFrameTransferStartFrameID  
     \_quickSpin, [104](#)  
 EventFrameTransferStartTimestamp  
     \_quickSpin, [104](#)  
 EventFrameTrigger  
     \_quickSpin, [104](#)  
 EventFrameTriggerFrameID  
     \_quickSpin, [105](#)  
 EventFrameTriggerTimestamp  
     \_quickSpin, [105](#)  
 EventLine0AnyEdge  
     \_quickSpin, [105](#)  
 EventLine0AnyEdgeFrameID  
     \_quickSpin, [105](#)  
 EventLine0AnyEdgeTimestamp  
     \_quickSpin, [105](#)  
 EventLine0FallingEdge  
     \_quickSpin, [105](#)  
 EventLine0FallingEdgeFrameID  
     \_quickSpin, [105](#)  
 EventLine0FallingEdgeTimestamp  
     \_quickSpin, [105](#)  
 EventLine0RisingEdge  
     \_quickSpin, [106](#)  
 EventLine0RisingEdgeFrameID  
     \_quickSpin, [106](#)  
 EventLine0RisingEdgeTimestamp  
     \_quickSpin, [106](#)  
 EventLine1AnyEdge  
     \_quickSpin, [106](#)  
 EventLine1AnyEdgeFrameID  
     \_quickSpin, [106](#)  
 EventLine1AnyEdgeTimestamp  
     \_quickSpin, [106](#)  
 EventLine1FallingEdge  
     \_quickSpin, [106](#)  
 EventLine1FallingEdgeFrameID  
     \_quickSpin, [106](#)  
 EventLine1FallingEdgeTimestamp  
     \_quickSpin, [107](#)  
 EventLine1RisingEdge  
     \_quickSpin, [107](#)  
 EventLine1RisingEdgeFrameID  
     \_quickSpin, [107](#)  
 EventLine1RisingEdgeTimestamp  
     \_quickSpin, [107](#)  
 EventLinkSpeedChange  
     \_quickSpin, [107](#)  
 EventLinkSpeedChangeFrameID  
     \_quickSpin, [107](#)  
 EventLinkSpeedChangeTimestamp  
     \_quickSpin, [107](#)  
 EventLinkTrigger0  
     \_quickSpin, [107](#)  
 EventLinkTrigger0FrameID  
     \_quickSpin, [108](#)  
 EventLinkTrigger0Timestamp  
     \_quickSpin, [108](#)  
 EventLinkTrigger1  
     \_quickSpin, [108](#)  
 EventLinkTrigger1FrameID  
     \_quickSpin, [108](#)  
 EventLinkTrigger1Timestamp  
     \_quickSpin, [108](#)  
 EventNotification  
     \_quickSpin, [108](#)  
 EventNotification\_Off  
     CameraDefsC.h, [254](#)  
 EventNotification\_On  
     CameraDefsC.h, [254](#)  
 EventSelector  
     \_quickSpin, [108](#)  
 EventSelector\_Error  
     CameraDefsC.h, [254](#)  
 EventSelector\_ExposureEnd  
     CameraDefsC.h, [254](#)  
 EventSelector\_SerialPortReceive  
     CameraDefsC.h, [254](#)  
 EventSequencerSetChange  
     \_quickSpin, [108](#)  
 EventSequencerSetChangeFrameID  
     \_quickSpin, [109](#)  
 EventSequencerSetChangeTimestamp  
     \_quickSpin, [109](#)  
 EventSerialData  
     \_quickSpin, [109](#)  
 EventSerialDataLength  
     \_quickSpin, [109](#)  
 EventSerialPortReceive  
     \_quickSpin, [109](#)  
 EventSerialPortReceiveTimestamp  
     \_quickSpin, [109](#)  
 EventSerialReceiveOverflow  
     \_quickSpin, [109](#)  
 EventStream0TransferBlockEnd  
     \_quickSpin, [109](#)  
 EventStream0TransferBlockEndFrameID  
     \_quickSpin, [110](#)  
 EventStream0TransferBlockEndTimestamp  
     \_quickSpin, [110](#)  
 EventStream0TransferBlockStart  
     \_quickSpin, [110](#)  
 EventStream0TransferBlockStartFrameID  
     \_quickSpin, [110](#)  
 EventStream0TransferBlockStartTimestamp  
     \_quickSpin, [110](#)  
 EventStream0TransferBlockTrigger  
     \_quickSpin, [110](#)  
 EventStream0TransferBlockTriggerFrameID  
     \_quickSpin, [110](#)

- EventStream0TransferBlockTriggerTimestamp
  - [\\_quickSpin, 110](#)
- EventStream0TransferBurstEnd
  - [\\_quickSpin, 111](#)
- EventStream0TransferBurstEndFrameID
  - [\\_quickSpin, 111](#)
- EventStream0TransferBurstEndTimestamp
  - [\\_quickSpin, 111](#)
- EventStream0TransferBurstStart
  - [\\_quickSpin, 111](#)
- EventStream0TransferBurstStartFrameID
  - [\\_quickSpin, 111](#)
- EventStream0TransferBurstStartTimestamp
  - [\\_quickSpin, 111](#)
- EventStream0TransferEnd
  - [\\_quickSpin, 111](#)
- EventStream0TransferEndFrameID
  - [\\_quickSpin, 111](#)
- EventStream0TransferEndTimestamp
  - [\\_quickSpin, 112](#)
- EventStream0TransferOverflow
  - [\\_quickSpin, 112](#)
- EventStream0TransferOverflowFrameID
  - [\\_quickSpin, 112](#)
- EventStream0TransferOverflowTimestamp
  - [\\_quickSpin, 112](#)
- EventStream0TransferPause
  - [\\_quickSpin, 112](#)
- EventStream0TransferPauseFrameID
  - [\\_quickSpin, 112](#)
- EventStream0TransferPauseTimestamp
  - [\\_quickSpin, 112](#)
- EventStream0TransferResume
  - [\\_quickSpin, 112](#)
- EventStream0TransferResumeFrameID
  - [\\_quickSpin, 113](#)
- EventStream0TransferResumeTimestamp
  - [\\_quickSpin, 113](#)
- EventStream0TransferStart
  - [\\_quickSpin, 113](#)
- EventStream0TransferStartFrameID
  - [\\_quickSpin, 113](#)
- EventStream0TransferStartTimestamp
  - [\\_quickSpin, 113](#)
- EventTest
  - [\\_quickSpin, 113](#)
- EventTestTimestamp
  - [\\_quickSpin, 113](#)
- EventTimer0End
  - [\\_quickSpin, 113](#)
- EventTimer0EndFrameID
  - [\\_quickSpin, 114](#)
- EventTimer0EndTimestamp
  - [\\_quickSpin, 114](#)
- EventTimer0Start
  - [\\_quickSpin, 114](#)
- EventTimer0StartFrameID
  - [\\_quickSpin, 114](#)
- EventTimer0StartTimestamp
  - [\\_quickSpin, 114](#)
- EventTimer1End
  - [\\_quickSpin, 114](#)
- EventTimer1EndFrameID
  - [\\_quickSpin, 114](#)
- EventTimer1EndTimestamp
  - [\\_quickSpin, 114](#)
- EventTimer1Start
  - [\\_quickSpin, 115](#)
- EventTimer1StartFrameID
  - [\\_quickSpin, 115](#)
- EventTimer1StartTimestamp
  - [\\_quickSpin, 115](#)
- Expert
  - SpinnakerGenApiDefsC.h, [463](#)
- ExposureActiveMode
  - [\\_quickSpin, 115](#)
- ExposureActiveMode\_AllPixels
  - CameraDefsC.h, [254](#)
- ExposureActiveMode\_AnyPixels
  - CameraDefsC.h, [254](#)
- ExposureActiveMode\_Line1
  - CameraDefsC.h, [254](#)
- ExposureAuto
  - [\\_quickSpin, 115](#)
- ExposureAuto\_Continuous
  - CameraDefsC.h, [255](#)
- ExposureAuto\_Off
  - CameraDefsC.h, [255](#)
- ExposureAuto\_Once
  - CameraDefsC.h, [255](#)
- ExposureMode
  - [\\_quickSpin, 115](#)
- ExposureMode\_Timed
  - CameraDefsC.h, [255](#)
- ExposureMode\_TriggerWidth
  - CameraDefsC.h, [255](#)
- ExposureTime
  - [\\_quickSpin, 115](#)
- ExposureTimeMode
  - [\\_quickSpin, 115](#)
- ExposureTimeMode\_Common
  - CameraDefsC.h, [255](#)
- ExposureTimeMode\_Individual
  - CameraDefsC.h, [255](#)
- ExposureTimeSelector
  - [\\_quickSpin, 116](#)
- ExposureTimeSelector\_Blue
  - CameraDefsC.h, [256](#)
- ExposureTimeSelector\_Common
  - CameraDefsC.h, [256](#)
- ExposureTimeSelector\_Cyan
  - CameraDefsC.h, [256](#)
- ExposureTimeSelector\_Green
  - CameraDefsC.h, [256](#)
- ExposureTimeSelector\_Infrared
  - CameraDefsC.h, [256](#)

- ExposureTimeSelector\_Magenta
  - CameraDefsC.h, [256](#)
- ExposureTimeSelector\_Red
  - CameraDefsC.h, [256](#)
- ExposureTimeSelector\_Stage1
  - CameraDefsC.h, [256](#)
- ExposureTimeSelector\_Stage2
  - CameraDefsC.h, [256](#)
- ExposureTimeSelector\_Ultraviolet
  - CameraDefsC.h, [256](#)
- ExposureTimeSelector\_Yellow
  - CameraDefsC.h, [256](#)
- FactoryReset
  - \_quickSpin, [116](#)
- False
  - SpinnakerDefsC.h, [414](#)
- FileAccessBuffer
  - \_quickSpin, [116](#)
- FileAccessLength
  - \_quickSpin, [116](#)
- FileAccessOffset
  - \_quickSpin, [116](#)
- FileOpenMode
  - \_quickSpin, [116](#)
- FileOpenMode\_Read
  - CameraDefsC.h, [256](#)
- FileOpenMode\_ReadWrite
  - CameraDefsC.h, [256](#)
- FileOpenMode\_Write
  - CameraDefsC.h, [256](#)
- FileOperationExecute
  - \_quickSpin, [116](#)
- FileOperationResult
  - \_quickSpin, [116](#)
- FileOperationSelector
  - \_quickSpin, [117](#)
- FileOperationSelector\_Close
  - CameraDefsC.h, [257](#)
- FileOperationSelector\_Delete
  - CameraDefsC.h, [257](#)
- FileOperationSelector\_Open
  - CameraDefsC.h, [257](#)
- FileOperationSelector\_Read
  - CameraDefsC.h, [257](#)
- FileOperationSelector\_Write
  - CameraDefsC.h, [257](#)
- FileOperationStatus
  - \_quickSpin, [117](#)
- FileOperationStatus\_Failure
  - CameraDefsC.h, [257](#)
- FileOperationStatus\_Overflow
  - CameraDefsC.h, [257](#)
- FileOperationStatus\_Success
  - CameraDefsC.h, [257](#)
- FileSelector
  - \_quickSpin, [117](#)
- FileSelector\_SerialPort0
  - CameraDefsC.h, [257](#)
- FileSelector\_UserFile1
  - CameraDefsC.h, [257](#)
- FileSelector\_UserSet0
  - CameraDefsC.h, [257](#)
- FileSelector\_UserSet1
  - CameraDefsC.h, [257](#)
- FileSelector\_UserSetDefault
  - CameraDefsC.h, [257](#)
- FileSize
  - \_quickSpin, [117](#)
- FilterDriverStatus
  - \_quickSpinTLInterface, [156](#)
- FilterDriverStatus\_Disabled
  - TransportLayerDefsC.h, [471](#)
- FilterDriverStatus\_Enabled
  - TransportLayerDefsC.h, [471](#)
- FilterDriverStatus\_NotSupported
  - TransportLayerDefsC.h, [471](#)
- fixedIncrement
  - SpinnakerGenApiDefsC.h, [458](#)
- FloatNode
  - SpinnakerGenApiDefsC.h, [461](#)
- fnAutomatic
  - SpinnakerGenApiDefsC.h, [458](#)
- fnFixed
  - SpinnakerGenApiDefsC.h, [458](#)
- fnScientific
  - SpinnakerGenApiDefsC.h, [458](#)
- frameRate
  - \_spinAVIOption, [170](#)
  - \_spinH264Option, [178](#)
  - \_spinMJPGOption, [182](#)
- FROM\_FILE\_EXT
  - SpinnakerDefsC.h, [410](#)
- Gain
  - \_quickSpin, [117](#)
- GainAuto
  - \_quickSpin, [117](#)
- GainAuto\_Continuous
  - CameraDefsC.h, [259](#)
- GainAuto\_Off
  - CameraDefsC.h, [259](#)
- GainAuto\_Once
  - CameraDefsC.h, [259](#)
- GainAutoBalance
  - \_quickSpin, [117](#)
- GainAutoBalance\_Continuous
  - CameraDefsC.h, [259](#)
- GainAutoBalance\_Off
  - CameraDefsC.h, [259](#)
- GainAutoBalance\_Once
  - CameraDefsC.h, [259](#)
- GainSelector
  - \_quickSpin, [117](#)
- GainSelector\_All
  - CameraDefsC.h, [259](#)
- Gamma
  - \_quickSpin, [118](#)

GammaEnable  
     \_quickSpin, 118  
 GENICAM\_ERR\_ACCESS  
     SpinnakerDefsC.h, 410  
 GENICAM\_ERR\_BAD\_ALLOCATION  
     SpinnakerDefsC.h, 410  
 GENICAM\_ERR\_DYNAMIC\_CAST  
     SpinnakerDefsC.h, 410  
 GENICAM\_ERR\_GENERIC  
     SpinnakerDefsC.h, 410  
 GENICAM\_ERR\_INVALID\_ARGUMENT  
     SpinnakerDefsC.h, 410  
 GENICAM\_ERR\_LOGICAL  
     SpinnakerDefsC.h, 410  
 GENICAM\_ERR\_OUT\_OF\_RANGE  
     SpinnakerDefsC.h, 410  
 GENICAM\_ERR\_PROPERTY  
     SpinnakerDefsC.h, 410  
 GENICAM\_ERR\_RUN\_TIME  
     SpinnakerDefsC.h, 410  
 GENICAM\_ERR\_TIMEOUT  
     SpinnakerDefsC.h, 410  
 GenICamXMLLocation  
     \_quickSpinTLDevice, 151  
 GenICamXMLLocation\_Device  
     TransportLayerDefsC.h, 472  
 GenICamXMLLocation\_Host  
     TransportLayerDefsC.h, 472  
 GenICamXMLPath  
     \_quickSpinTLDevice, 151  
 GenTLFNCVersionMajor  
     \_quickSpinTLSystem, 166  
 GenTLFNCVersionMinor  
     \_quickSpinTLSystem, 167  
 GenTLFNCVersionSubMinor  
     \_quickSpinTLSystem, 167  
 GenTLVersionMajor  
     \_quickSpinTLSystem, 167  
 GenTLVersionMinor  
     \_quickSpinTLSystem, 167  
 GEV  
     SpinnakerGenApiDefsC.h, 462  
 GevActionDeviceKey  
     \_quickSpinTLInterface, 157  
 GevActionGroupKey  
     \_quickSpinTLInterface, 157  
 GevActionGroupMask  
     \_quickSpinTLInterface, 157  
 GevActionTime  
     \_quickSpinTLInterface, 157  
 GevActiveLinkCount  
     \_quickSpin, 118  
 GevCCP  
     \_quickSpin, 118  
     \_quickSpinTLDevice, 151  
 GevCCP\_ControlAccess  
     CameraDefsC.h, 260  
 GevCCP\_EnumEntry\_GevCCP\_ControlAccess  
     TransportLayerDefsC.h, 472  
 GevCCP\_EnumEntry\_GevCCP\_ExclusiveAccess  
     TransportLayerDefsC.h, 472  
 GevCCP\_EnumEntry\_GevCCP\_OpenAccess  
     TransportLayerDefsC.h, 472  
 GevCCP\_ExclusiveAccess  
     CameraDefsC.h, 260  
 GevCCP\_OpenAccess  
     CameraDefsC.h, 260  
 GevCurrentDefaultGateway  
     \_quickSpin, 118  
 GevCurrentIPAddress  
     \_quickSpin, 118  
 GevCurrentIPConfigurationDHCP  
     \_quickSpin, 118  
 GevCurrentIPConfigurationLLA  
     \_quickSpin, 118  
 GevCurrentIPConfigurationPersistentIP  
     \_quickSpin, 119  
 GevCurrentPhysicalLinkConfiguration  
     \_quickSpin, 119  
 GevCurrentPhysicalLinkConfiguration\_DynamicLAG  
     CameraDefsC.h, 260  
 GevCurrentPhysicalLinkConfiguration\_MultiLink  
     CameraDefsC.h, 260  
 GevCurrentPhysicalLinkConfiguration\_SingleLink  
     CameraDefsC.h, 260  
 GevCurrentPhysicalLinkConfiguration\_StaticLAG  
     CameraDefsC.h, 260  
 GevCurrentSubnetMask  
     \_quickSpin, 119  
 GevDeviceAutoForceIP  
     \_quickSpinTLDevice, 152  
     \_quickSpinTLInterface, 157  
 GevDeviceDiscoverMaximumPacketSize  
     \_quickSpinTLDevice, 152  
 GevDeviceForceGateway  
     \_quickSpinTLDevice, 152  
     \_quickSpinTLInterface, 157  
 GevDeviceForceIP  
     \_quickSpinTLDevice, 152  
     \_quickSpinTLInterface, 157  
 GevDeviceForceIPAddress  
     \_quickSpinTLDevice, 152  
     \_quickSpinTLInterface, 157  
 GevDeviceForceSubnetMask  
     \_quickSpinTLDevice, 152  
     \_quickSpinTLInterface, 158  
 GevDeviceGateway  
     \_quickSpinTLDevice, 152  
     \_quickSpinTLInterface, 158  
 GevDeviceIPAddress  
     \_quickSpinTLDevice, 152  
     \_quickSpinTLInterface, 158  
 GevDeviceIsWrongSubnet  
     \_quickSpinTLDevice, 153  
 GevDeviceMACAddress  
     \_quickSpinTLDevice, 153



- [\\_quickSpinTLInterface](#), 158
- [GevDeviceMaximumPacketSize](#)
- [\\_quickSpinTLDevice](#), 153
- [GevDeviceMaximumRetryCount](#)
- [\\_quickSpinTLDevice](#), 153
- [GevDeviceModelsBigEndian](#)
- [\\_quickSpinTLDevice](#), 153
- [GevDevicePort](#)
- [\\_quickSpinTLDevice](#), 153
- [GevDeviceReadAndWriteTimeout](#)
- [\\_quickSpinTLDevice](#), 153
- [GevDeviceSubnetMask](#)
- [\\_quickSpinTLDevice](#), 153
- [\\_quickSpinTLInterface](#), 158
- [GevDiscoveryAckDelay](#)
- [\\_quickSpin](#), 119
- [GevFailedPacketCount](#)
- [\\_quickSpinTLStream](#), 162
- [GevFirstURL](#)
- [\\_quickSpin](#), 119
- [GevGVCPExtendedStatusCodes](#)
- [\\_quickSpin](#), 119
- [GevGVCPExtendedStatusCodesSelector](#)
- [\\_quickSpin](#), 119
- [GevGVCPExtendedStatusCodesSelector\\_Version1\\_1](#)
- [CameraDefsC.h](#), 260
- [GevGVCPExtendedStatusCodesSelector\\_Version2\\_0](#)
- [CameraDefsC.h](#), 260
- [GevGVCPHeartbeatDisable](#)
- [\\_quickSpin](#), 119
- [GevGVCPPendingAck](#)
- [\\_quickSpin](#), 120
- [GevGVCPPendingTimeout](#)
- [\\_quickSpin](#), 120
- [GevGVSPExtendedIDMode](#)
- [\\_quickSpin](#), 120
- [GevGVSPExtendedIDMode\\_Off](#)
- [CameraDefsC.h](#), 261
- [GevGVSPExtendedIDMode\\_On](#)
- [CameraDefsC.h](#), 261
- [GevHeartbeatTimeout](#)
- [\\_quickSpin](#), 120
- [GevIEEE1588](#)
- [\\_quickSpin](#), 120
- [GevIEEE1588ClockAccuracy](#)
- [\\_quickSpin](#), 120
- [GevIEEE1588ClockAccuracy\\_Unknown](#)
- [CameraDefsC.h](#), 261
- [GevIEEE1588Mode](#)
- [\\_quickSpin](#), 120
- [GevIEEE1588Mode\\_Auto](#)
- [CameraDefsC.h](#), 261
- [GevIEEE1588Mode\\_SlaveOnly](#)
- [CameraDefsC.h](#), 261
- [GevIEEE1588Status](#)
- [\\_quickSpin](#), 120
- [GevIEEE1588Status\\_Disabled](#)
- [CameraDefsC.h](#), 261
- [GevIEEE1588Status\\_Faulty](#)
- [CameraDefsC.h](#), 261
- [GevIEEE1588Status\\_Initializing](#)
- [CameraDefsC.h](#), 261
- [GevIEEE1588Status\\_Listening](#)
- [CameraDefsC.h](#), 262
- [GevIEEE1588Status\\_Master](#)
- [CameraDefsC.h](#), 262
- [GevIEEE1588Status\\_Passive](#)
- [CameraDefsC.h](#), 262
- [GevIEEE1588Status\\_PreMaster](#)
- [CameraDefsC.h](#), 262
- [GevIEEE1588Status\\_Slave](#)
- [CameraDefsC.h](#), 262
- [GevIEEE1588Status\\_Uncalibrated](#)
- [CameraDefsC.h](#), 262
- [GevInterfaceDefaultGateway](#)
- [\\_quickSpinTLSystem](#), 167
- [GevInterfaceDefaultIPAddress](#)
- [\\_quickSpinTLSystem](#), 167
- [GevInterfaceDefaultSubnetMask](#)
- [\\_quickSpinTLSystem](#), 167
- [GevInterfaceGateway](#)
- [\\_quickSpinTLInterface](#), 158
- [GevInterfaceGatewaySelector](#)
- [\\_quickSpinTLInterface](#), 158
- [GevInterfaceMACAddress](#)
- [\\_quickSpinTLInterface](#), 158
- [\\_quickSpinTLSystem](#), 167
- [GevInterfaceMTU](#)
- [\\_quickSpinTLInterface](#), 159
- [GevInterfaceReceiveLinkSpeed](#)
- [\\_quickSpinTLInterface](#), 159
- [GevInterfaceSelector](#)
- [\\_quickSpin](#), 121
- [GevInterfaceSubnetIPAddress](#)
- [\\_quickSpinTLInterface](#), 159
- [GevInterfaceSubnetMask](#)
- [\\_quickSpinTLInterface](#), 159
- [GevInterfaceSubnetSelector](#)
- [\\_quickSpinTLInterface](#), 159
- [GevInterfaceTransmitLinkSpeed](#)
- [\\_quickSpinTLInterface](#), 159
- [GevIPConfigurationStatus](#)
- [\\_quickSpin](#), 121
- [GevIPConfigurationStatus\\_DHCP](#)
- [CameraDefsC.h](#), 262
- [GevIPConfigurationStatus\\_ForceIP](#)
- [CameraDefsC.h](#), 262
- [GevIPConfigurationStatus\\_LLA](#)
- [CameraDefsC.h](#), 262
- [GevIPConfigurationStatus\\_None](#)
- [CameraDefsC.h](#), 262
- [GevIPConfigurationStatus\\_PersistentIP](#)
- [CameraDefsC.h](#), 262
- [GevMACAddress](#)
- [\\_quickSpin](#), 121
- [GevMaximumNumberResendRequests](#)



- [\\_quickSpinTLStream, 162](#)
- GevMCDA
  - [\\_quickSpin, 121](#)
- GevMCPHostPort
  - [\\_quickSpin, 121](#)
- GevMCRC
  - [\\_quickSpin, 121](#)
- GevMCSP
  - [\\_quickSpin, 121](#)
- GevMCTT
  - [\\_quickSpin, 121](#)
- GevNumberOfInterfaces
  - [\\_quickSpin, 122](#)
- GevPacketResendMode
  - [\\_quickSpinTLStream, 162](#)
- GevPacketResendTimeout
  - [\\_quickSpinTLStream, 163](#)
- GevPAUSEFrameReception
  - [\\_quickSpin, 122](#)
- GevPAUSEFrameTransmission
  - [\\_quickSpin, 122](#)
- GevPersistentDefaultGateway
  - [\\_quickSpin, 122](#)
- GevPersistentIPAddress
  - [\\_quickSpin, 122](#)
- GevPersistentSubnetMask
  - [\\_quickSpin, 122](#)
- GevPhysicalLinkConfiguration
  - [\\_quickSpin, 122](#)
- GevPhysicalLinkConfiguration\_DynamicLAG
  - [CameraDefsC.h, 262](#)
- GevPhysicalLinkConfiguration\_MultiLink
  - [CameraDefsC.h, 262](#)
- GevPhysicalLinkConfiguration\_SingleLink
  - [CameraDefsC.h, 262](#)
- GevPhysicalLinkConfiguration\_StaticLAG
  - [CameraDefsC.h, 262](#)
- GevPrimaryApplicationIPAddress
  - [\\_quickSpin, 122](#)
- GevPrimaryApplicationSocket
  - [\\_quickSpin, 123](#)
- GevPrimaryApplicationSwitchoverKey
  - [\\_quickSpin, 123](#)
- GevResendPacketCount
  - [\\_quickSpinTLStream, 163](#)
- GevResendRequestCount
  - [\\_quickSpinTLStream, 163](#)
- GevSCCFGAllInTransmission
  - [\\_quickSpin, 123](#)
- GevSCCFGExtendedChunkData
  - [\\_quickSpin, 123](#)
- GevSCCFGPacketResendDestination
  - [\\_quickSpin, 123](#)
- GevSCCFGUnconditionalStreaming
  - [\\_quickSpin, 123](#)
- GevSCDA
  - [\\_quickSpin, 123](#)
- GevSCPD
  - [\\_quickSpin, 123](#)
- GevSCPDirection
  - [\\_quickSpin, 124](#)
- GevSCPHostPort
  - [\\_quickSpin, 124](#)
- GevSCPInterfaceIndex
  - [\\_quickSpin, 124](#)
- GevSCPSBigEndian
  - [\\_quickSpin, 124](#)
- GevSCPSDoNotFragment
  - [\\_quickSpin, 124](#)
- GevSCPSFireTestPacket
  - [\\_quickSpin, 124](#)
- GevSCPSPacketSize
  - [\\_quickSpin, 124](#)
- GevSCSP
  - [\\_quickSpin, 124](#)
- GevSCZoneConfigurationLock
  - [\\_quickSpin, 125](#)
- GevSCZoneCount
  - [\\_quickSpin, 125](#)
- GevSCZoneDirectionAll
  - [\\_quickSpin, 125](#)
- GevSecondURL
  - [\\_quickSpin, 125](#)
- GevStreamChannelSelector
  - [\\_quickSpin, 125](#)
- GevSupportedOption
  - [\\_quickSpin, 125](#)
- GevSupportedOptionSelector
  - [\\_quickSpin, 125](#)
- GevSupportedOptionSelector\_Action
  - [CameraDefsC.h, 263](#)
- GevSupportedOptionSelector\_CCPAApplicationSocket
  - [CameraDefsC.h, 263](#)
- GevSupportedOptionSelector\_CommandsConcatenation
  - [CameraDefsC.h, 263](#)
- GevSupportedOptionSelector\_DiscoveryAckDelay
  - [CameraDefsC.h, 263](#)
- GevSupportedOptionSelector\_DiscoveryAckDelayWritable
  - [CameraDefsC.h, 263](#)
- GevSupportedOptionSelector\_Event
  - [CameraDefsC.h, 263](#)
- GevSupportedOptionSelector\_EventData
  - [CameraDefsC.h, 263](#)
- GevSupportedOptionSelector\_ExtendedStatusCodes
  - [CameraDefsC.h, 263](#)
- GevSupportedOptionSelector\_HeartbeatDisable
  - [CameraDefsC.h, 263](#)
- GevSupportedOptionSelector\_IPConfigurationDHCP
  - [CameraDefsC.h, 263](#)
- GevSupportedOptionSelector\_IPConfigurationLLA
  - [CameraDefsC.h, 263](#)
- GevSupportedOptionSelector\_IPConfigurationPersistentIP
  - [CameraDefsC.h, 263](#)
- GevSupportedOptionSelector\_LinkSpeed
  - [CameraDefsC.h, 263](#)
- GevSupportedOptionSelector\_ManifestTable

- CameraDefsC.h, [263](#)
- GevSupportedOptionSelector\_MessageChannelSourceSocket
  - CameraDefsC.h, [263](#)
- GevSupportedOptionSelector\_PacketResend
  - CameraDefsC.h, [263](#)
- GevSupportedOptionSelector\_PendingAck
  - CameraDefsC.h, [263](#)
- GevSupportedOptionSelector\_SerialNumber
  - CameraDefsC.h, [263](#)
- GevSupportedOptionSelector\_StreamChannelSourceSocket
  - CameraDefsC.h, [263](#)
- GevSupportedOptionSelector\_TestData
  - CameraDefsC.h, [263](#)
- GevSupportedOptionSelector\_UserDefinedName
  - CameraDefsC.h, [263](#)
- GevSupportedOptionSelector\_WriteMem
  - CameraDefsC.h, [263](#)
- GevTimestampTickFrequency
  - \_quickSpin, [125](#)
- GevTotalPacketCount
  - \_quickSpinTLStream, [163](#)
- GevVersionMajor
  - \_quickSpinTLDevice, [154](#)
  - \_quickSpinTLSystem, [168](#)
- GevVersionMinor
  - \_quickSpinTLDevice, [154](#)
  - \_quickSpinTLSystem, [168](#)
- GREEN
  - SpinnakerDefsC.h, [413](#)
- GREY
  - SpinnakerDefsC.h, [413](#)
- GUIXMLLocation
  - \_quickSpinTLDevice, [154](#)
- GUIXMLLocation\_Device
  - TransportLayerDefsC.h, [472](#)
- GUIXMLLocation\_Host
  - TransportLayerDefsC.h, [472](#)
- GuiXmlManifestAddress
  - \_quickSpin, [126](#)
- GUIXMLPath
  - \_quickSpinTLDevice, [154](#)
- Guru
  - SpinnakerGenApiDefsC.h, [463](#)
- Height
  - \_quickSpin, [126](#)
- height
  - \_spinH264Option, [178](#)
- HeightMax
  - \_quickSpin, [126](#)
- HexNumber
  - SpinnakerGenApiDefsC.h, [461](#)
- HostAdapterDriverVersion
  - \_quickSpinTLInterface, [159](#)
- HostAdapterName
  - \_quickSpinTLInterface, [159](#)
- HostAdapterVendor
  - \_quickSpinTLInterface, [160](#)
- HQ\_LINEAR
  - SpinnakerDefsC.h, [409](#)
  - SpinnakerDefsC.h, [413](#)
- IBoolean Access, [38](#)
- ICategory Access, [40](#)
- ICommand Access, [39](#)
- idFrom
  - SpinnakerGenApiDefsC.h, [459](#)
- idNone
  - SpinnakerGenApiDefsC.h, [459](#)
- idTo
  - SpinnakerGenApiDefsC.h, [459](#)
- IEnumEntry Access, [37](#)
- IEnumeration Access, [36](#)
- IFloat Access, [35](#)
- IIDC
  - SpinnakerGenApiDefsC.h, [462](#)
- Integer Access, [34](#)
- Image Access, [19](#)
- IMAGE\_CHUNK\_DATA\_INVALID
  - SpinnakerDefsC.h, [411](#)
- IMAGE\_CRC\_CHECK\_FAILED
  - SpinnakerDefsC.h, [411](#)
- IMAGE\_DATA\_INCOMPLETE
  - SpinnakerDefsC.h, [411](#)
- IMAGE\_DATA\_OVERFLOW
  - SpinnakerDefsC.h, [411](#)
- IMAGE\_FILE\_FORMAT\_FORCE\_32BITS
  - SpinnakerDefsC.h, [411](#)
- IMAGE\_INFO\_INCONSISTENT
  - SpinnakerDefsC.h, [411](#)
- IMAGE\_LEADER\_BUFFER\_SIZE\_INCONSISTENT
  - SpinnakerDefsC.h, [411](#)
- IMAGE\_MISSING\_LEADER
  - SpinnakerDefsC.h, [411](#)
- IMAGE\_MISSING\_PACKETS
  - SpinnakerDefsC.h, [411](#)
- IMAGE\_MISSING\_TRAILER
  - SpinnakerDefsC.h, [411](#)
- IMAGE\_NO\_ERROR
  - SpinnakerDefsC.h, [411](#)
- IMAGE\_NO\_SYSTEM\_RESOURCES
  - SpinnakerDefsC.h, [411](#)
- IMAGE\_PACKETID\_INCONSISTENT
  - SpinnakerDefsC.h, [411](#)
- IMAGE\_TRAILER\_BUFFER\_SIZE\_INCONSISTENT
  - SpinnakerDefsC.h, [411](#)
- IMAGE\_UNKNOWN\_ERROR
  - SpinnakerDefsC.h, [411](#)
- ImageComponentEnable
  - \_quickSpin, [126](#)
- ImageComponentSelector
  - \_quickSpin, [126](#)
- ImageComponentSelector\_Color
  - CameraDefsC.h, [263](#)
- ImageComponentSelector\_Confidence
  - CameraDefsC.h, [264](#)
- ImageComponentSelector\_Disparity

- CameraDefsC.h, [264](#)
- ImageComponentSelector\_Infrared
  - CameraDefsC.h, [263](#)
- ImageComponentSelector\_Intensity
  - CameraDefsC.h, [263](#)
- ImageComponentSelector\_Range
  - CameraDefsC.h, [263](#)
- ImageComponentSelector\_Scatter
  - CameraDefsC.h, [264](#)
- ImageComponentSelector\_Ultraviolet
  - CameraDefsC.h, [263](#)
- ImageCompressionBitrate
  - \_quickSpin, [126](#)
- ImageCompressionJPEGFormatOption
  - \_quickSpin, [126](#)
- ImageCompressionJPEGFormatOption\_BaselineOptimized
  - CameraDefsC.h, [264](#)
- ImageCompressionJPEGFormatOption\_BaselineStandardInterface Access, [17](#)
  - CameraDefsC.h, [264](#)
- ImageCompressionJPEGFormatOption\_Lossless
  - CameraDefsC.h, [264](#)
- ImageCompressionJPEGFormatOption\_Progressive
  - CameraDefsC.h, [264](#)
- ImageCompressionMode
  - \_quickSpin, [126](#)
- ImageCompressionMode\_Lossless
  - CameraDefsC.h, [265](#)
- ImageCompressionMode\_Off
  - CameraDefsC.h, [265](#)
- ImageCompressionQuality
  - \_quickSpin, [127](#)
- ImageCompressionRateOption
  - \_quickSpin, [127](#)
- ImageCompressionRateOption\_FixBitrate
  - CameraDefsC.h, [265](#)
- ImageCompressionRateOption\_FixQuality
  - CameraDefsC.h, [265](#)
- ImageStatistics Access, [21](#)
- include/spinc/CameraDefsC.h, [187](#)
- include/spinc/ChunkDataDefC.h, [303](#)
- include/spinc/QuickSpinC.h, [304](#)
- include/spinc/QuickSpinDefsC.h, [306](#)
- include/spinc/SpinnakerC.h, [308](#)
- include/spinc/SpinnakerDefsC.h, [400](#)
- include/spinc/SpinnakerGenApiC.h, [414](#)
- include/spinc/SpinnakerGenApiDefsC.h, [453](#)
- include/spinc/SpinnakerPlatformC.h, [464](#)
- include/spinc/SpinVideoC.h, [465](#)
- include/spinc/TransportLayerDefsC.h, [467](#)
- include/spinc/TransportLayerDeviceC.h, [475](#)
- include/spinc/TransportLayerInterfaceC.h, [476](#)
- include/spinc/TransportLayerStreamC.h, [476](#)
- include/spinc/TransportLayerSystemC.h, [477](#)
- IncompatibleDeviceCount
  - \_quickSpinTLInterface, [160](#)
- IncompatibleDeviceID
  - \_quickSpinTLInterface, [160](#)
- IncompatibleDeviceModelName
  - \_quickSpinTLInterface, [160](#)
- IncompatibleDeviceSelector
  - \_quickSpinTLInterface, [160](#)
- IncompatibleDeviceVendorName
  - \_quickSpinTLInterface, [160](#)
- IncompatibleGevDeviceIPAddress
  - \_quickSpinTLInterface, [160](#)
- IncompatibleGevDeviceMACAddress
  - \_quickSpinTLInterface, [160](#)
- IncompatibleGevDeviceSubnetMask
  - \_quickSpinTLInterface, [161](#)
- Increasing
  - SpinnakerGenApiDefsC.h, [462](#)
- indexedColor\_8bit
  - \_spinBMPOption, [171](#)
- IntegerNode
  - SpinnakerGenApiDefsC.h, [461](#)
- Interface Access, [17](#)
  - InterfaceDisplayName
    - \_quickSpinTLInterface, [161](#)
    - \_quickSpinTLSystem, [168](#)
  - InterfaceID
    - \_quickSpinTLInterface, [161](#)
    - \_quickSpinTLSystem, [168](#)
  - InterfaceList Access, [15](#)
  - InterfaceSelector
    - \_quickSpinTLSystem, [168](#)
  - InterfaceType
    - \_quickSpinTLInterface, [161](#)
  - InterfaceType\_CameraLink
    - TransportLayerDefsC.h, [472](#)
  - InterfaceType\_CameraLinkHS
    - TransportLayerDefsC.h, [473](#)
  - InterfaceType\_CoaxPress
    - TransportLayerDefsC.h, [473](#)
  - InterfaceType\_Custom
    - TransportLayerDefsC.h, [473](#)
  - InterfaceType\_GigEVision
    - TransportLayerDefsC.h, [472](#)
  - InterfaceType\_USB3Vision
    - TransportLayerDefsC.h, [473](#)
  - InterfaceUpdateList
    - \_quickSpinTLSystem, [168](#)
- interlaced
  - \_spinPNGOption, [184](#)
- intfIBase
  - SpinnakerGenApiDefsC.h, [459](#)
- intfIBoolean
  - SpinnakerGenApiDefsC.h, [459](#)
- intfICategory
  - SpinnakerGenApiDefsC.h, [459](#)
- intfICommand
  - SpinnakerGenApiDefsC.h, [459](#)
- intfIEnumEntry
  - SpinnakerGenApiDefsC.h, [459](#)
- intfIEnumeration
  - SpinnakerGenApiDefsC.h, [459](#)
- intfIFloat

- SpinnakerGenApiDefsC.h, [459](#)
- intfInteger
  - SpinnakerGenApiDefsC.h, [459](#)
- intfIPort
  - SpinnakerGenApiDefsC.h, [460](#)
- intfRegister
  - SpinnakerGenApiDefsC.h, [459](#)
- intfString
  - SpinnakerGenApiDefsC.h, [459](#)
- intfValue
  - SpinnakerGenApiDefsC.h, [459](#)
- Invisible
  - SpinnakerGenApiDefsC.h, [463](#)
- IPP
  - SpinnakerDefsC.h, [409](#)
- IPv4Address
  - SpinnakerGenApiDefsC.h, [461](#)
- IRegister Access, [41](#)
- IspEnable
  - \_quickSpin, [127](#)
- IValue Access, [32](#)
- JPEG
  - SpinnakerDefsC.h, [410](#)
- JPEG2000
  - SpinnakerDefsC.h, [410](#)
- JPG
  - SpinnakerDefsC.h, [413](#)
- LIGHTNESS
  - SpinnakerDefsC.h, [413](#)
- Linear
  - SpinnakerGenApiDefsC.h, [461](#)
- LineFilterWidth
  - \_quickSpin, [127](#)
- LineFormat
  - \_quickSpin, [127](#)
- LineFormat\_LVDS
  - CameraDefsC.h, [265](#)
- LineFormat\_NoConnect
  - CameraDefsC.h, [265](#)
- LineFormat\_OpenDrain
  - CameraDefsC.h, [265](#)
- LineFormat\_OptoCoupled
  - CameraDefsC.h, [265](#)
- LineFormat\_RS422
  - CameraDefsC.h, [265](#)
- LineFormat\_TriState
  - CameraDefsC.h, [265](#)
- LineFormat\_TTL
  - CameraDefsC.h, [265](#)
- LineInputFilterSelector
  - \_quickSpin, [127](#)
- LineInputFilterSelector\_Debounce
  - CameraDefsC.h, [266](#)
- LineInputFilterSelector\_Deglitch
  - CameraDefsC.h, [266](#)
- LineInverter
  - \_quickSpin, [127](#)
- LineMode
  - \_quickSpin, [127](#)
- LineMode\_Input
  - CameraDefsC.h, [266](#)
- LineMode\_Output
  - CameraDefsC.h, [266](#)
- LinePitch
  - \_quickSpin, [128](#)
- LineSelector
  - \_quickSpin, [128](#)
- LineSelector\_Line0
  - CameraDefsC.h, [266](#)
- LineSelector\_Line1
  - CameraDefsC.h, [266](#)
- LineSelector\_Line2
  - CameraDefsC.h, [266](#)
- LineSelector\_Line3
  - CameraDefsC.h, [266](#)
- LineSource
  - \_quickSpin, [128](#)
- LineSource\_AllPixel
  - CameraDefsC.h, [267](#)
- LineSource\_AnyPixel
  - CameraDefsC.h, [267](#)
- LineSource\_Counter0Active
  - CameraDefsC.h, [267](#)
- LineSource\_Counter1Active
  - CameraDefsC.h, [267](#)
- LineSource\_ExposureActive
  - CameraDefsC.h, [267](#)
- LineSource\_FrameTriggerWait
  - CameraDefsC.h, [267](#)
- LineSource\_Line0
  - CameraDefsC.h, [267](#)
- LineSource\_Line1
  - CameraDefsC.h, [267](#)
- LineSource\_Line2
  - CameraDefsC.h, [267](#)
- LineSource\_Line3
  - CameraDefsC.h, [267](#)
- LineSource\_LogicBlock0
  - CameraDefsC.h, [267](#)
- LineSource\_LogicBlock1
  - CameraDefsC.h, [267](#)
- LineSource\_Off
  - CameraDefsC.h, [267](#)
- LineSource\_PPSSignal
  - CameraDefsC.h, [267](#)
- LineSource\_SerialPort0
  - CameraDefsC.h, [267](#)
- LineSource\_UserOutput0
  - CameraDefsC.h, [267](#)
- LineSource\_UserOutput1
  - CameraDefsC.h, [267](#)
- LineSource\_UserOutput2
  - CameraDefsC.h, [267](#)
- LineSource\_UserOutput3
  - CameraDefsC.h, [267](#)

LineStatus  
     \_quickSpin, 128  
 LineStatusAll  
     \_quickSpin, 128  
 LinkErrorCount  
     \_quickSpin, 128  
 LinkUptime  
     \_quickSpin, 128  
 listIncrement  
     SpinnakerGenApiDefsC.h, 458  
 LittleEndian  
     SpinnakerGenApiDefsC.h, 458  
 LOG\_LEVEL\_ALERT  
     SpinnakerDefsC.h, 412  
 LOG\_LEVEL\_CRIT  
     SpinnakerDefsC.h, 412  
 LOG\_LEVEL\_DEBUG  
     SpinnakerDefsC.h, 412  
 LOG\_LEVEL\_ERROR  
     SpinnakerDefsC.h, 412  
 LOG\_LEVEL\_FATAL  
     SpinnakerDefsC.h, 412  
 LOG\_LEVEL\_INFO  
     SpinnakerDefsC.h, 412  
 LOG\_LEVEL\_NOTICE  
     SpinnakerDefsC.h, 412  
 LOG\_LEVEL\_NOTSET  
     SpinnakerDefsC.h, 412  
 LOG\_LEVEL\_OFF  
     SpinnakerDefsC.h, 412  
 LOG\_LEVEL\_WARN  
     SpinnakerDefsC.h, 412  
 Logarithmic  
     SpinnakerGenApiDefsC.h, 461  
 Logging Event Data Access, 22  
 LogicBlockLUTInputActivation  
     \_quickSpin, 128  
 LogicBlockLUTInputActivation\_AnyEdge  
     CameraDefsC.h, 267  
 LogicBlockLUTInputActivation\_FallingEdge  
     CameraDefsC.h, 267  
 LogicBlockLUTInputActivation\_LevelHigh  
     CameraDefsC.h, 267  
 LogicBlockLUTInputActivation\_LevelLow  
     CameraDefsC.h, 267  
 LogicBlockLUTInputActivation\_RisingEdge  
     CameraDefsC.h, 267  
 LogicBlockLUTInputSelector  
     \_quickSpin, 129  
 LogicBlockLUTInputSelector\_Input0  
     CameraDefsC.h, 268  
 LogicBlockLUTInputSelector\_Input1  
     CameraDefsC.h, 268  
 LogicBlockLUTInputSelector\_Input2  
     CameraDefsC.h, 268  
 LogicBlockLUTInputSelector\_Input3  
     CameraDefsC.h, 268  
 LogicBlockLUTInputSource  
     \_quickSpin, 129  
 LogicBlockLUTInputSource\_AcquisitionActive  
     CameraDefsC.h, 268  
 LogicBlockLUTInputSource\_Counter0End  
     CameraDefsC.h, 268  
 LogicBlockLUTInputSource\_Counter0Start  
     CameraDefsC.h, 268  
 LogicBlockLUTInputSource\_Counter1End  
     CameraDefsC.h, 268  
 LogicBlockLUTInputSource\_Counter1Start  
     CameraDefsC.h, 268  
 LogicBlockLUTInputSource\_ExposureEnd  
     CameraDefsC.h, 268  
 LogicBlockLUTInputSource\_ExposureStart  
     CameraDefsC.h, 268  
 LogicBlockLUTInputSource\_FrameTriggerWait  
     CameraDefsC.h, 268  
 LogicBlockLUTInputSource\_Line0  
     CameraDefsC.h, 268  
 LogicBlockLUTInputSource\_Line1  
     CameraDefsC.h, 268  
 LogicBlockLUTInputSource\_Line2  
     CameraDefsC.h, 268  
 LogicBlockLUTInputSource\_Line3  
     CameraDefsC.h, 268  
 LogicBlockLUTInputSource\_LogicBlock0  
     CameraDefsC.h, 268  
 LogicBlockLUTInputSource\_LogicBlock1  
     CameraDefsC.h, 268  
 LogicBlockLUTInputSource\_UserOutput0  
     CameraDefsC.h, 268  
 LogicBlockLUTInputSource\_UserOutput1  
     CameraDefsC.h, 268  
 LogicBlockLUTInputSource\_UserOutput2  
     CameraDefsC.h, 268  
 LogicBlockLUTInputSource\_UserOutput3  
     CameraDefsC.h, 268  
 LogicBlockLUTInputSource\_Zero  
     CameraDefsC.h, 268  
 LogicBlockLUTOutputValue  
     \_quickSpin, 129  
 LogicBlockLUTOutputValueAll  
     \_quickSpin, 129  
 LogicBlockLUTRowIndex  
     \_quickSpin, 129  
 LogicBlockLUTSelector  
     \_quickSpin, 129  
 LogicBlockLUTSelector\_Enable  
     CameraDefsC.h, 269  
 LogicBlockLUTSelector\_Value  
     CameraDefsC.h, 269  
 LogicBlockSelector  
     \_quickSpin, 129  
 LogicBlockSelector\_LogicBlock0  
     CameraDefsC.h, 269  
 LogicBlockSelector\_LogicBlock1  
     CameraDefsC.h, 269  
 LUTEnable

- [\\_quickSpin, 129](#)
- LUTIndex
  - [\\_quickSpin, 130](#)
- LUTSelector
  - [\\_quickSpin, 130](#)
- LUTSelector\_LUT1
  - [CameraDefsC.h, 269](#)
- LUTValue
  - [\\_quickSpin, 130](#)
- LUTValueAll
  - [\\_quickSpin, 130](#)
- LZW
  - [SpinnakerDefsC.h, 413](#)
- m\_blackLevel
  - [\\_spinChunkData, 172](#)
- m\_counterValue
  - [\\_spinChunkData, 172](#)
- m\_cRC
  - [\\_spinChunkData, 173](#)
- m\_encoderValue
  - [\\_spinChunkData, 173](#)
- m\_exposureEndLineStatusAll
  - [\\_spinChunkData, 173](#)
- m\_exposureTime
  - [\\_spinChunkData, 173](#)
- m\_frameID
  - [\\_spinChunkData, 173](#)
- m\_gain
  - [\\_spinChunkData, 173](#)
- m\_height
  - [\\_spinChunkData, 173](#)
- m\_image
  - [\\_spinChunkData, 173](#)
- m\_inferenceConfidence
  - [\\_spinChunkData, 174](#)
- m\_inferenceFrameID
  - [\\_spinChunkData, 174](#)
- m\_inferenceResult
  - [\\_spinChunkData, 174](#)
- m\_linePitch
  - [\\_spinChunkData, 174](#)
- m\_lineStatusAll
  - [\\_spinChunkData, 174](#)
- m\_offsetX
  - [\\_spinChunkData, 174](#)
- m\_offsetY
  - [\\_spinChunkData, 174](#)
- m\_partSelector
  - [\\_spinChunkData, 174](#)
- m\_pixelDynamicRangeMax
  - [\\_spinChunkData, 175](#)
- m\_pixelDynamicRangeMin
  - [\\_spinChunkData, 175](#)
- m\_scan3dAxisMax
  - [\\_spinChunkData, 175](#)
- m\_scan3dAxisMin
  - [\\_spinChunkData, 175](#)
- m\_scan3dCoordinateOffset
  - [\\_spinChunkData, 175](#)
- m\_scan3dCoordinateReferenceValue
  - [\\_spinChunkData, 175](#)
- m\_scan3dCoordinateScale
  - [\\_spinChunkData, 175](#)
- m\_scan3dInvalidDataValue
  - [\\_spinChunkData, 175](#)
- m\_scan3dTransformValue
  - [\\_spinChunkData, 176](#)
- m\_scanLineSelector
  - [\\_spinChunkData, 176](#)
- m\_sequencerSetActive
  - [\\_spinChunkData, 176](#)
- m\_serialDataLength
  - [\\_spinChunkData, 176](#)
- m\_streamChannelID
  - [\\_spinChunkData, 176](#)
- m\_timerValue
  - [\\_spinChunkData, 176](#)
- m\_timestamp
  - [\\_spinChunkData, 176](#)
- m\_timestampLatchValue
  - [\\_spinChunkData, 176](#)
- m\_transferBlockID
  - [\\_spinChunkData, 177](#)
- m\_transferQueueCurrentBlockCount
  - [\\_spinChunkData, 177](#)
- m\_width
  - [\\_spinChunkData, 177](#)
- MACAddress
  - [SpinnakerGenApiDefsC.h, 461](#)
- major
  - [\\_spinLibraryVersion, 181](#)
- MaxDeviceResetTime
  - [\\_quickSpin, 130](#)
- minor
  - [\\_spinLibraryVersion, 181](#)
- NA
  - [SpinnakerGenApiDefsC.h, 457](#)
- NEAREST\_NEIGHBOR
  - [SpinnakerDefsC.h, 409](#)
- NEAREST\_NEIGHBOR\_AVG
  - [SpinnakerDefsC.h, 409](#)
- NI
  - [SpinnakerGenApiDefsC.h, 457](#)
- No
  - [SpinnakerGenApiDefsC.h, 464](#)
- NO\_COLOR\_PROCESSING
  - [SpinnakerDefsC.h, 409](#)
- NoCache
  - [SpinnakerGenApiDefsC.h, 457](#)
- Node Access, [31](#)
- Node Map Access, [30](#)
- noIncrement
  - [SpinnakerGenApiDefsC.h, 458](#)
- NONE
  - [SpinnakerDefsC.h, 413](#)
- None



- SpinnakerGenApiDefsC.h, [462](#)
- NUM\_ACQUISITIONMODE
  - CameraDefsC.h, [220](#)
- NUM\_ACQUISITIONSTATUSSELECTION
  - CameraDefsC.h, [220](#)
- NUM\_ACTIONUNCONDITIONALMODE
  - CameraDefsC.h, [220](#)
- NUM\_ADCBITDEPTH
  - CameraDefsC.h, [221](#)
- NUM\_AUTOALGORITHMSELECTION
  - CameraDefsC.h, [221](#)
- NUM\_AUTOEXPOSURECONTROLPRIORITY
  - CameraDefsC.h, [221](#)
- NUM\_AUTOEXPOSURELIGHTINGMODE
  - CameraDefsC.h, [222](#)
- NUM\_AUTOEXPOSUREMETERINGMODE
  - CameraDefsC.h, [222](#)
- NUM\_AUTOEXPOSURETARGETGREYVALUEAUTO
  - CameraDefsC.h, [222](#)
- NUM\_BALANCERATIOSELECTION
  - CameraDefsC.h, [223](#)
- NUM\_BALANCEWHITEAUTO
  - CameraDefsC.h, [223](#)
- NUM\_BALANCEWHITEAUTOPROFILE
  - CameraDefsC.h, [223](#)
- NUM\_BINNINGHORIZONTALMODE
  - CameraDefsC.h, [224](#)
- NUM\_BINNINGSELECTION
  - CameraDefsC.h, [224](#)
- NUM\_BINNINGVERTICALMODE
  - CameraDefsC.h, [224](#)
- NUM\_BLACKLEVELAUTO
  - CameraDefsC.h, [225](#)
- NUM\_BLACKLEVELAUTOBALANCE
  - CameraDefsC.h, [225](#)
- NUM\_BLACKLEVELSELECTION
  - CameraDefsC.h, [225](#)
- NUM\_CHUNKBLACKLEVELSELECTION
  - CameraDefsC.h, [226](#)
- NUM\_CHUNKCOUNTERSELECTION
  - CameraDefsC.h, [226](#)
- NUM\_CHUNKENCODERSELECTION
  - CameraDefsC.h, [226](#)
- NUM\_CHUNKENCODERSTATUS
  - CameraDefsC.h, [227](#)
- NUM\_CHUNKEXPOSURETIMESELECTION
  - CameraDefsC.h, [227](#)
- NUM\_CHUNKGAINSELECTION
  - CameraDefsC.h, [227](#)
- NUM\_CHUNKIMAGECOMPONENT
  - CameraDefsC.h, [228](#)
- NUM\_CHUNKPIXELFORMAT
  - CameraDefsC.h, [228](#)
- NUM\_CHUNKREGIONID
  - CameraDefsC.h, [229](#)
- NUM\_CHUNKSCAN3DCOORDINATEREFERENCESELECTION
  - CameraDefsC.h, [229](#)
- NUM\_CHUNKSCAN3DCOORDINATESELECTION
  - CameraDefsC.h, [229](#)
- NUM\_CHUNKSCAN3DCOORDINATESYSTEM
  - CameraDefsC.h, [230](#)
- NUM\_CHUNKSCAN3DCOORDINATESYSTEMREFERENCE
  - CameraDefsC.h, [230](#)
- NUM\_CHUNKSCAN3DCOORDINATETRANSFORMSELECTION
  - CameraDefsC.h, [230](#)
- NUM\_CHUNKSCAN3DDISTANCEUNIT
  - CameraDefsC.h, [231](#)
- NUM\_CHUNKSCAN3DOUTPUTMODE
  - CameraDefsC.h, [232](#)
- NUM\_CHUNKSELECTION
  - CameraDefsC.h, [232](#)
- NUM\_CHUNKSOURCEID
  - CameraDefsC.h, [233](#)
- NUM\_CHUNKTIMERSELECTION
  - CameraDefsC.h, [233](#)
- NUM\_CHUNKTRANSFERSTREAMID
  - CameraDefsC.h, [233](#)
- NUM\_CLCONFIGURATION
  - CameraDefsC.h, [234](#)
- NUM\_CLTIMESLOTSCOUNT
  - CameraDefsC.h, [234](#)
- NUM\_COLORTRANSFORMATIONSELECTION
  - CameraDefsC.h, [234](#)
- NUM\_COLORTRANSFORMATIONVALUESELECTION
  - CameraDefsC.h, [235](#)
- NUM\_COUNTEREVENTACTIVATION
  - CameraDefsC.h, [235](#)
- NUM\_COUNTEREVENTSOURCE
  - CameraDefsC.h, [236](#)
- NUM\_COUNTERRESETACTIVATION
  - CameraDefsC.h, [236](#)
- NUM\_COUNTERRESETSOURCE
  - CameraDefsC.h, [237](#)
- NUM\_COUNTERSELECTION
  - CameraDefsC.h, [237](#)
- NUM\_COUNTERSTATUS
  - CameraDefsC.h, [237](#)
- NUM\_COUNTERTRIGGERACTIVATION
  - CameraDefsC.h, [238](#)
- NUM\_COUNTERTRIGGERSOURCE
  - CameraDefsC.h, [238](#)
- NUM\_CXPCONNECTIONTESTMODE
  - CameraDefsC.h, [239](#)
- NUM\_CXPLINKCONFIGURATION
  - CameraDefsC.h, [240](#)
- NUM\_CXPLINKCONFIGURATIONPREFERRED
  - CameraDefsC.h, [241](#)
- NUM\_CXPLINKCONFIGURATIONSTATUS
  - CameraDefsC.h, [242](#)
- NUM\_CXPPOCXPSTATUS
  - CameraDefsC.h, [242](#)
- NUM\_DECIMATIONHORIZONTALMODE
  - CameraDefsC.h, [242](#)
- NUM\_DECIMATIONSELECTION
  - CameraDefsC.h, [243](#)
- NUM\_DECIMATIONVERTICALMODE

- CameraDefsC.h, [243](#)
- NUM\_DEFECTCORRECTIONMODE
  - CameraDefsC.h, [243](#)
- NUM\_DEINTERLACING
  - CameraDefsC.h, [244](#)
- NUM\_DEVICECHARACTERSET
  - CameraDefsC.h, [244](#)
- NUM\_DEVICECLOCKSELECTOR
  - CameraDefsC.h, [244](#)
- NUM\_DEVICECONNECTIONSTATUS
  - CameraDefsC.h, [244](#)
- NUM\_DEVICEINDICATORMODE
  - CameraDefsC.h, [245](#)
- NUM\_DEVICELINKHEARTBEATMODE
  - CameraDefsC.h, [245](#)
- NUM\_DEVICELINKTHROUGHPUTLIMITMODE
  - CameraDefsC.h, [245](#)
- NUM\_DEVICEPOWERSUPPLYSELECTOR
  - CameraDefsC.h, [246](#)
- NUM\_DEVICEREGISTERSENDIANNES
  - CameraDefsC.h, [246](#)
- NUM\_DEVICESCANTYPE
  - CameraDefsC.h, [246](#)
- NUM\_DEVICESERIALPORTBAUDRATE
  - CameraDefsC.h, [247](#)
- NUM\_DEVICESERIALPORTSELECTOR
  - CameraDefsC.h, [247](#)
- NUM\_DEVICESTREAMCHANNELENDIANNES
  - CameraDefsC.h, [247](#)
- NUM\_DEVICESTREAMCHANNELTYPE
  - CameraDefsC.h, [248](#)
- NUM\_DEVICETAPGEOMETRY
  - CameraDefsC.h, [249](#)
- NUM\_DEVICETEMPERATURESELECTOR
  - CameraDefsC.h, [249](#)
- NUM\_DEVICETLTYPE
  - CameraDefsC.h, [249](#)
- NUM\_DEVICETYPE
  - CameraDefsC.h, [250](#)
- NUM\_ENCODERMODE
  - CameraDefsC.h, [250](#)
- NUM\_ENCODEROUTPUTMODE
  - CameraDefsC.h, [251](#)
- NUM\_ENCODERRESETACTIVATION
  - CameraDefsC.h, [251](#)
- NUM\_ENCODERRESETSOURCE
  - CameraDefsC.h, [252](#)
- NUM\_ENCODERSELECTOR
  - CameraDefsC.h, [252](#)
- NUM\_ENCODERSOURCEA
  - CameraDefsC.h, [253](#)
- NUM\_ENCODERSOURCEB
  - CameraDefsC.h, [253](#)
- NUM\_ENCODERSTATUS
  - CameraDefsC.h, [253](#)
- NUM\_EVENTNOTIFICATION
  - CameraDefsC.h, [254](#)
- NUM\_EVENTSELECTOR
  - CameraDefsC.h, [254](#)
- NUM\_EXPOSUREACTIVEMODE
  - CameraDefsC.h, [254](#)
- NUM\_EXPOSUREAUTO
  - CameraDefsC.h, [255](#)
- NUM\_EXPOSUREMODE
  - CameraDefsC.h, [255](#)
- NUM\_EXPOSURETIMEMODE
  - CameraDefsC.h, [255](#)
- NUM\_EXPOSURETIMESELECTOR
  - CameraDefsC.h, [256](#)
- NUM\_FILEOPENMODE
  - CameraDefsC.h, [256](#)
- NUM\_FILEOPERATIONSELECTOR
  - CameraDefsC.h, [257](#)
- NUM\_FILEOPERATIONSTATUS
  - CameraDefsC.h, [257](#)
- NUM\_FILESELECTOR
  - CameraDefsC.h, [257](#)
- NUM\_GAINAUTO
  - CameraDefsC.h, [259](#)
- NUM\_GAINAUTOBALANCE
  - CameraDefsC.h, [259](#)
- NUM\_GAINSELECTOR
  - CameraDefsC.h, [259](#)
- NUM\_GEVCCP
  - CameraDefsC.h, [260](#)
- NUM\_GEVCURRENTPHYSICALLINKCONFIGURATION
  - CameraDefsC.h, [260](#)
- NUM\_GEVGVCPEXTENDEDSTATUSCODESSELECTOR
  - CameraDefsC.h, [260](#)
- NUM\_GEVGVSPEXTENDEDIDMODE
  - CameraDefsC.h, [261](#)
- NUM\_GEVEEEE1588CLOCKACCURACY
  - CameraDefsC.h, [261](#)
- NUM\_GEVEEEE1588MODE
  - CameraDefsC.h, [261](#)
- NUM\_GEVEEEE1588STATUS
  - CameraDefsC.h, [262](#)
- NUM\_GEVIPCONFIGURATIONSTATUS
  - CameraDefsC.h, [262](#)
- NUM\_GEVPHYSICALLINKCONFIGURATION
  - CameraDefsC.h, [262](#)
- NUM\_GEVSUPPORTEDOPTIONSELECTOR
  - CameraDefsC.h, [263](#)
- NUM\_IMAGECOMPONENTSELECTOR
  - CameraDefsC.h, [264](#)
- NUM\_IMAGECOMPRESSIONJPEGFORMATOPTION
  - CameraDefsC.h, [264](#)
- NUM\_IMAGECOMPRESSIONMODE
  - CameraDefsC.h, [265](#)
- NUM\_IMAGECOMPRESSIONRATEOPTION
  - CameraDefsC.h, [265](#)
- NUM\_LINEFORMAT
  - CameraDefsC.h, [265](#)
- NUM\_LINEINPUTFILTERSELECTOR
  - CameraDefsC.h, [266](#)
- NUM\_LINEMODE



- CameraDefsC.h, [266](#)
- NUM\_LINESELECTOR
  - CameraDefsC.h, [266](#)
- NUM\_LINESOURCE
  - CameraDefsC.h, [267](#)
- NUM\_LOGICBLOCKLUTINPUTACTIVATION
  - CameraDefsC.h, [267](#)
- NUM\_LOGICBLOCKLUTINPUTSELECTOR
  - CameraDefsC.h, [268](#)
- NUM\_LOGICBLOCKLUTINPUTSOURCE
  - CameraDefsC.h, [268](#)
- NUM\_LOGICBLOCKLUTSELECTOR
  - CameraDefsC.h, [269](#)
- NUM\_LOGICBLOCKSELECTOR
  - CameraDefsC.h, [269](#)
- NUM\_LUTSELECTOR
  - CameraDefsC.h, [269](#)
- NUM\_PIXELCOLORFILTER
  - CameraDefsC.h, [270](#)
- NUM\_PIXELFORMAT
  - CameraDefsC.h, [275](#)
- NUM\_PIXELFORMATINFOSELECTOR
  - CameraDefsC.h, [281](#)
- NUM\_PIXELSIZE
  - CameraDefsC.h, [282](#)
- NUM\_REGIONDESTINATION
  - CameraDefsC.h, [282](#)
- NUM\_REGIONMODE
  - CameraDefsC.h, [282](#)
- NUM\_REGIONSELECTOR
  - CameraDefsC.h, [283](#)
- NUM\_RGBTRANSFORMLIGHTSOURCE
  - CameraDefsC.h, [283](#)
- NUM\_SCAN3DCOORDINATEREFERENCESELECTOR
  - CameraDefsC.h, [284](#)
- NUM\_SCAN3DCOORDINATESELECTOR
  - CameraDefsC.h, [284](#)
- NUM\_SCAN3DCOORDINATESYSTEM
  - CameraDefsC.h, [284](#)
- NUM\_SCAN3DCOORDINATESYSTEMREFERENCE
  - CameraDefsC.h, [285](#)
- NUM\_SCAN3DCOORDINATETRANSFORMSELECTOR
  - CameraDefsC.h, [285](#)
- NUM\_SCAN3DDISTANCEUNIT
  - CameraDefsC.h, [285](#)
- NUM\_SCAN3DOUTPUTMODE
  - CameraDefsC.h, [286](#)
- NUM\_SENSORDIGITIZATIONTAPS
  - CameraDefsC.h, [287](#)
- NUM\_SENSORSHUTTERMODE
  - CameraDefsC.h, [287](#)
- NUM\_SENSORTAPS
  - CameraDefsC.h, [287](#)
- NUM\_SEQUENCERCONFIGURATIONMODE
  - CameraDefsC.h, [288](#)
- NUM\_SEQUENCERCONFIGURATIONVALID
  - CameraDefsC.h, [288](#)
- NUM\_SEQUENCERMODE
  - CameraDefsC.h, [288](#)
- NUM\_SEQUENCERSETVALID
  - CameraDefsC.h, [289](#)
- NUM\_SEQUENCERTRIGGERACTIVATION
  - CameraDefsC.h, [289](#)
- NUM\_SEQUENCERTRIGGERSOURCE
  - CameraDefsC.h, [289](#)
- NUM\_SERIALPORTBAUDRATE
  - CameraDefsC.h, [290](#)
- NUM\_SERIALPORTPARITY
  - CameraDefsC.h, [290](#)
- NUM\_SERIALPORTSELECTOR
  - CameraDefsC.h, [290](#)
- NUM\_SERIALPORTSOURCE
  - CameraDefsC.h, [291](#)
- NUM\_SERIALPORTSTOPBITS
  - CameraDefsC.h, [291](#)
- NUM\_SOFTWARESIGNALSELECTOR
  - CameraDefsC.h, [291](#)
- NUM\_SOURCESELECTOR
  - CameraDefsC.h, [292](#)
- NUM\_STATISTICS\_CHANNELS
  - SpinnakerDefsC.h, [413](#)
- NUM\_TESTPATTERN
  - CameraDefsC.h, [292](#)
- NUM\_TESTPATTERNGENERATORSELECTOR
  - CameraDefsC.h, [292](#)
- NUM\_TIMERSELECTOR
  - CameraDefsC.h, [293](#)
- NUM\_TIMERSTATUS
  - CameraDefsC.h, [293](#)
- NUM\_TIMERTRIGGERACTIVATION
  - CameraDefsC.h, [293](#)
- NUM\_TIMERTRIGGERSOURCE
  - CameraDefsC.h, [295](#)
- NUM\_TRANSFERCOMPONENTSELECTOR
  - CameraDefsC.h, [295](#)
- NUM\_TRANSFERCONTROLMODE
  - CameraDefsC.h, [296](#)
- NUM\_TRANSFEROPERATIONMODE
  - CameraDefsC.h, [296](#)
- NUM\_TRANSFERQUEUEMODE
  - CameraDefsC.h, [296](#)
- NUM\_TRANSFERSELECTOR
  - CameraDefsC.h, [296](#)
- NUM\_TRANSFERSTATUSSELECTOR
  - CameraDefsC.h, [297](#)
- NUM\_TRANSFERTRIGGERACTIVATION
  - CameraDefsC.h, [297](#)
- NUM\_TRANSFERTRIGGERMODE
  - CameraDefsC.h, [298](#)
- NUM\_TRANSFERTRIGGERSELECTOR
  - CameraDefsC.h, [298](#)
- NUM\_TRANSFERTRIGGERSOURCE
  - CameraDefsC.h, [299](#)
- NUM\_TRIGGERACTIVATION
  - CameraDefsC.h, [299](#)
- NUM\_TRIGGERMODE

- CameraDefsC.h, [300](#)
- NUM\_TRIGGEROVERLAP
  - CameraDefsC.h, [300](#)
- NUM\_TRIGGERSELECTOR
  - CameraDefsC.h, [300](#)
- NUM\_TRIGGERSOURCE
  - CameraDefsC.h, [301](#)
- NUM\_USEROUTPUTSELECTOR
  - CameraDefsC.h, [301](#)
- NUM\_USERSETDEFAULT
  - CameraDefsC.h, [302](#)
- NUM\_USERSETSELECTOR
  - CameraDefsC.h, [302](#)
- NUM\_WHITECLIPSELECTOR
  - CameraDefsC.h, [302](#)
- NUMDEVICEACCESSSTATUS
  - TransportLayerDefsC.h, [469](#)
- NUMDEVICECURRENTSPEED
  - TransportLayerDefsC.h, [469](#)
- NUMDEVICEENDIANESSMECHANISM
  - TransportLayerDefsC.h, [471](#)
- NUMDEVICETYPE
  - TransportLayerDefsC.h, [471](#)
- NUMFILTERDRIVERSTATUS
  - TransportLayerDefsC.h, [471](#)
- NUMGENICAMXMLLOCATION
  - TransportLayerDefsC.h, [472](#)
- NUMGEVCCP
  - TransportLayerDefsC.h, [472](#)
- NUMGUIXMLLOCATION
  - TransportLayerDefsC.h, [472](#)
- NUMINTERFACETYPE
  - TransportLayerDefsC.h, [473](#)
- NUMPOESTATUS
  - TransportLayerDefsC.h, [473](#)
- NUMSTREAMBUFFERCOUNTMODE
  - TransportLayerDefsC.h, [473](#)
- NUMSTREAMBUFFERHANDLINGMODE
  - TransportLayerDefsC.h, [474](#)
- NUMSTREAMTYPE
  - TransportLayerDefsC.h, [474](#)
- NUMTLTYPE
  - TransportLayerDefsC.h, [475](#)
- OffsetX
  - \_quickSpin, [130](#)
- OffsetY
  - \_quickSpin, [130](#)
- PACKBITS
  - SpinnakerDefsC.h, [413](#)
- PacketResendRequestCount
  - \_quickSpin, [130](#)
- PAYLOAD\_TYPE\_CHUNK\_DATA
  - SpinnakerDefsC.h, [412](#)
- PAYLOAD\_TYPE\_CHUNK\_ONLY
  - SpinnakerDefsC.h, [412](#)
- PAYLOAD\_TYPE\_CUSTOM\_ID
  - SpinnakerDefsC.h, [412](#)
- PAYLOAD\_TYPE\_DEVICE\_SPECIFIC
  - SpinnakerDefsC.h, [412](#)
- PAYLOAD\_TYPE\_EXTENDED\_CHUNK
  - SpinnakerDefsC.h, [412](#)
- PAYLOAD\_TYPE\_FILE
  - SpinnakerDefsC.h, [412](#)
- PAYLOAD\_TYPE\_H264
  - SpinnakerDefsC.h, [412](#)
- PAYLOAD\_TYPE\_IMAGE
  - SpinnakerDefsC.h, [412](#)
- PAYLOAD\_TYPE\_JPEG
  - SpinnakerDefsC.h, [412](#)
- PAYLOAD\_TYPE\_JPEG2000
  - SpinnakerDefsC.h, [412](#)
- PAYLOAD\_TYPE\_MULTI\_PART
  - SpinnakerDefsC.h, [412](#)
- PAYLOAD\_TYPE\_RAW\_DATA
  - SpinnakerDefsC.h, [412](#)
- PAYLOAD\_TYPE\_UNKNOWN
  - SpinnakerDefsC.h, [412](#)
- PayloadSize
  - \_quickSpin, [131](#)
- PGM
  - SpinnakerDefsC.h, [410](#)
- PixelColorFilter
  - \_quickSpin, [131](#)
- PixelColorFilter\_BayerBG
  - CameraDefsC.h, [270](#)
- PixelColorFilter\_BayerGB
  - CameraDefsC.h, [270](#)
- PixelColorFilter\_BayerGR
  - CameraDefsC.h, [270](#)
- PixelColorFilter\_BayerRG
  - CameraDefsC.h, [269](#)
- PixelColorFilter\_None
  - CameraDefsC.h, [269](#)
- PixelDynamicRangeMax
  - \_quickSpin, [131](#)
- PixelDynamicRangeMin
  - \_quickSpin, [131](#)
- PixelFormat
  - \_quickSpin, [131](#)
- PixelFormat\_B10
  - CameraDefsC.h, [272](#)
- PixelFormat\_B12
  - CameraDefsC.h, [272](#)
- PixelFormat\_B12\_Jpeg
  - CameraDefsC.h, [275](#)
- PixelFormat\_B16
  - CameraDefsC.h, [272](#)
- PixelFormat\_B8
  - CameraDefsC.h, [272](#)
- PixelFormat\_BayerBG10
  - CameraDefsC.h, [271](#)
- PixelFormat\_BayerBG10p
  - CameraDefsC.h, [271](#)
- PixelFormat\_BayerBG10Packed
  - CameraDefsC.h, [271](#)

PixelFormat\_BayerBG12  
CameraDefsC.h, [271](#)

PixelFormat\_BayerBG12p  
CameraDefsC.h, [270](#)

PixelFormat\_BayerBG12Packed  
CameraDefsC.h, [270](#)

PixelFormat\_BayerBG16  
CameraDefsC.h, [270](#)

PixelFormat\_BayerBG8  
CameraDefsC.h, [270](#)

PixelFormat\_BayerGB10  
CameraDefsC.h, [271](#)

PixelFormat\_BayerGB10p  
CameraDefsC.h, [271](#)

PixelFormat\_BayerGB10Packed  
CameraDefsC.h, [271](#)

PixelFormat\_BayerGB12  
CameraDefsC.h, [271](#)

PixelFormat\_BayerGB12p  
CameraDefsC.h, [270](#)

PixelFormat\_BayerGB12Packed  
CameraDefsC.h, [270](#)

PixelFormat\_BayerGB16  
CameraDefsC.h, [270](#)

PixelFormat\_BayerGB8  
CameraDefsC.h, [270](#)

PixelFormat\_BayerGR10  
CameraDefsC.h, [271](#)

PixelFormat\_BayerGR10p  
CameraDefsC.h, [271](#)

PixelFormat\_BayerGR10Packed  
CameraDefsC.h, [270](#)

PixelFormat\_BayerGR12  
CameraDefsC.h, [271](#)

PixelFormat\_BayerGR12p  
CameraDefsC.h, [270](#)

PixelFormat\_BayerGR12Packed  
CameraDefsC.h, [270](#)

PixelFormat\_BayerGR16  
CameraDefsC.h, [270](#)

PixelFormat\_BayerGR8  
CameraDefsC.h, [270](#)

PixelFormat\_BayerRG10  
CameraDefsC.h, [271](#)

PixelFormat\_BayerRG10p  
CameraDefsC.h, [271](#)

PixelFormat\_BayerRG10Packed  
CameraDefsC.h, [270](#)

PixelFormat\_BayerRG12  
CameraDefsC.h, [271](#)

PixelFormat\_BayerRG12p  
CameraDefsC.h, [270](#)

PixelFormat\_BayerRG12Packed  
CameraDefsC.h, [270](#)

PixelFormat\_BayerRG16  
CameraDefsC.h, [270](#)

PixelFormat\_BayerRG8  
CameraDefsC.h, [270](#)

PixelFormat\_BayerRGPolarized10p  
CameraDefsC.h, [275](#)

PixelFormat\_BayerRGPolarized12p  
CameraDefsC.h, [275](#)

PixelFormat\_BayerRGPolarized16  
CameraDefsC.h, [275](#)

PixelFormat\_BayerRGPolarized8  
CameraDefsC.h, [275](#)

PixelFormat\_BGR10  
CameraDefsC.h, [272](#)

PixelFormat\_BGR10p  
CameraDefsC.h, [272](#)

PixelFormat\_BGR12  
CameraDefsC.h, [272](#)

PixelFormat\_BGR12p  
CameraDefsC.h, [272](#)

PixelFormat\_BGR14  
CameraDefsC.h, [272](#)

PixelFormat\_BGR16  
CameraDefsC.h, [272](#)

PixelFormat\_BGR565p  
CameraDefsC.h, [272](#)

PixelFormat\_BGR8  
CameraDefsC.h, [270](#)

PixelFormat\_BGRa10  
CameraDefsC.h, [272](#)

PixelFormat\_BGRa10p  
CameraDefsC.h, [272](#)

PixelFormat\_BGRa12  
CameraDefsC.h, [272](#)

PixelFormat\_BGRa12p  
CameraDefsC.h, [272](#)

PixelFormat\_BGRa14  
CameraDefsC.h, [272](#)

PixelFormat\_BGRa16  
CameraDefsC.h, [272](#)

PixelFormat\_BGRa8  
CameraDefsC.h, [270](#)

PixelFormat\_BiColorBGRG10  
CameraDefsC.h, [273](#)

PixelFormat\_BiColorBGRG10p  
CameraDefsC.h, [273](#)

PixelFormat\_BiColorBGRG12  
CameraDefsC.h, [273](#)

PixelFormat\_BiColorBGRG12p  
CameraDefsC.h, [273](#)

PixelFormat\_BiColorBGRG8  
CameraDefsC.h, [273](#)

PixelFormat\_BiColorRGBG10  
CameraDefsC.h, [273](#)

PixelFormat\_BiColorRGBG10p  
CameraDefsC.h, [273](#)

PixelFormat\_BiColorRGBG12  
CameraDefsC.h, [273](#)

PixelFormat\_BiColorRGBG12p  
CameraDefsC.h, [273](#)

PixelFormat\_BiColorRGBG8  
CameraDefsC.h, [273](#)

- PixelFormat\_Confidence1
  - CameraDefsC.h, [273](#)
- PixelFormat\_Confidence16
  - CameraDefsC.h, [273](#)
- PixelFormat\_Confidence1p
  - CameraDefsC.h, [273](#)
- PixelFormat\_Confidence32f
  - CameraDefsC.h, [273](#)
- PixelFormat\_Confidence8
  - CameraDefsC.h, [273](#)
- PixelFormat\_Coord3D\_A10p
  - CameraDefsC.h, [273](#)
- PixelFormat\_Coord3D\_A12p
  - CameraDefsC.h, [273](#)
- PixelFormat\_Coord3D\_A16
  - CameraDefsC.h, [273](#)
- PixelFormat\_Coord3D\_A32f
  - CameraDefsC.h, [273](#)
- PixelFormat\_Coord3D\_A8
  - CameraDefsC.h, [273](#)
- PixelFormat\_Coord3D\_ABC10p
  - CameraDefsC.h, [272](#)
- PixelFormat\_Coord3D\_ABC10p\_Planar
  - CameraDefsC.h, [272](#)
- PixelFormat\_Coord3D\_ABC12p
  - CameraDefsC.h, [272](#)
- PixelFormat\_Coord3D\_ABC12p\_Planar
  - CameraDefsC.h, [272](#)
- PixelFormat\_Coord3D\_ABC16
  - CameraDefsC.h, [272](#)
- PixelFormat\_Coord3D\_ABC16\_Planar
  - CameraDefsC.h, [272](#)
- PixelFormat\_Coord3D\_ABC32f
  - CameraDefsC.h, [272](#)
- PixelFormat\_Coord3D\_ABC32f\_Planar
  - CameraDefsC.h, [272](#)
- PixelFormat\_Coord3D\_ABC8
  - CameraDefsC.h, [272](#)
- PixelFormat\_Coord3D\_ABC8\_Planar
  - CameraDefsC.h, [272](#)
- PixelFormat\_Coord3D\_AC10p
  - CameraDefsC.h, [272](#)
- PixelFormat\_Coord3D\_AC10p\_Planar
  - CameraDefsC.h, [272](#)
- PixelFormat\_Coord3D\_AC12p
  - CameraDefsC.h, [272](#)
- PixelFormat\_Coord3D\_AC12p\_Planar
  - CameraDefsC.h, [272](#)
- PixelFormat\_Coord3D\_AC16
  - CameraDefsC.h, [272](#)
- PixelFormat\_Coord3D\_AC16\_Planar
  - CameraDefsC.h, [272](#)
- PixelFormat\_Coord3D\_AC32f
  - CameraDefsC.h, [272](#)
- PixelFormat\_Coord3D\_AC32f\_Planar
  - CameraDefsC.h, [272](#)
- PixelFormat\_Coord3D\_AC8
  - CameraDefsC.h, [272](#)
- PixelFormat\_Coord3D\_AC8\_Planar
  - CameraDefsC.h, [272](#)
- PixelFormat\_G10
  - CameraDefsC.h, [272](#)
- PixelFormat\_G12
  - CameraDefsC.h, [272](#)
- PixelFormat\_G16
  - CameraDefsC.h, [272](#)
- PixelFormat\_G8
  - CameraDefsC.h, [272](#)
- PixelFormat\_GB12\_Jpeg
  - CameraDefsC.h, [275](#)
- PixelFormat\_GR12\_Jpeg
  - CameraDefsC.h, [275](#)
- PixelFormat\_JPEGColor8
  - CameraDefsC.h, [275](#)
- PixelFormat\_JPEGMono8
  - CameraDefsC.h, [275](#)
- PixelFormat\_LLCBayerRG8
  - CameraDefsC.h, [275](#)
- PixelFormat\_LLCMono8
  - CameraDefsC.h, [275](#)
- PixelFormat\_Mono10
  - CameraDefsC.h, [271](#)
- PixelFormat\_Mono10p
  - CameraDefsC.h, [271](#)
- PixelFormat\_Mono10Packed
  - CameraDefsC.h, [270](#)
- PixelFormat\_Mono12
  - CameraDefsC.h, [271](#)
- PixelFormat\_Mono12p
  - CameraDefsC.h, [270](#)
- PixelFormat\_Mono12Packed
  - CameraDefsC.h, [270](#)
- PixelFormat\_Mono14
  - CameraDefsC.h, [271](#)
- PixelFormat\_Mono16
  - CameraDefsC.h, [270](#)

- PixelFormat\_Mono16s
  - CameraDefsC.h, [271](#)
- PixelFormat\_Mono1p
  - CameraDefsC.h, [271](#)
- PixelFormat\_Mono2p
  - CameraDefsC.h, [271](#)
- PixelFormat\_Mono32f
  - CameraDefsC.h, [271](#)
- PixelFormat\_Mono4p
  - CameraDefsC.h, [271](#)
- PixelFormat\_Mono8
  - CameraDefsC.h, [270](#)
- PixelFormat\_Mono8s
  - CameraDefsC.h, [271](#)
- PixelFormat\_Polarized10p
  - CameraDefsC.h, [275](#)
- PixelFormat\_Polarized12p
  - CameraDefsC.h, [275](#)
- PixelFormat\_Polarized16
  - CameraDefsC.h, [275](#)
- PixelFormat\_Polarized8
  - CameraDefsC.h, [275](#)
- PixelFormat\_R10
  - CameraDefsC.h, [272](#)
- PixelFormat\_R12
  - CameraDefsC.h, [272](#)
- PixelFormat\_R12\_Jpeg
  - CameraDefsC.h, [275](#)
- PixelFormat\_R16
  - CameraDefsC.h, [272](#)
- PixelFormat\_R8
  - CameraDefsC.h, [272](#)
- PixelFormat\_Raw16
  - CameraDefsC.h, [275](#)
- PixelFormat\_Raw8
  - CameraDefsC.h, [275](#)
- PixelFormat\_RGB10
  - CameraDefsC.h, [271](#)
- PixelFormat\_RGB10\_Planar
  - CameraDefsC.h, [271](#)
- PixelFormat\_RGB10p
  - CameraDefsC.h, [271](#)
- PixelFormat\_RGB10p32
  - CameraDefsC.h, [271](#)
- PixelFormat\_RGB12
  - CameraDefsC.h, [271](#)
- PixelFormat\_RGB12\_Planar
  - CameraDefsC.h, [271](#)
- PixelFormat\_RGB12p
  - CameraDefsC.h, [271](#)
- PixelFormat\_RGB14
  - CameraDefsC.h, [271](#)
- PixelFormat\_RGB16
  - CameraDefsC.h, [271](#)
- PixelFormat\_RGB16\_Planar
  - CameraDefsC.h, [271](#)
- PixelFormat\_RGB16s
  - CameraDefsC.h, [271](#)
- PixelFormat\_RGB32f
  - CameraDefsC.h, [271](#)
- PixelFormat\_RGB565p
  - CameraDefsC.h, [272](#)
- PixelFormat\_RGB8
  - CameraDefsC.h, [271](#)
- PixelFormat\_RGB8\_Planar
  - CameraDefsC.h, [271](#)
- PixelFormat\_RGB8Packed
  - CameraDefsC.h, [270](#)
- PixelFormat\_RGBa10
  - CameraDefsC.h, [271](#)
- PixelFormat\_RGBa10p
  - CameraDefsC.h, [271](#)
- PixelFormat\_RGBa12
  - CameraDefsC.h, [271](#)
- PixelFormat\_RGBa12p
  - CameraDefsC.h, [271](#)
- PixelFormat\_RGBa14
  - CameraDefsC.h, [271](#)
- PixelFormat\_RGBa16
  - CameraDefsC.h, [271](#)
- PixelFormat\_RGBa32f
  - CameraDefsC.h, [272](#)
- PixelFormat\_RGBa8
  - CameraDefsC.h, [271](#)
- PixelFormat\_SCF1WBWG10
  - CameraDefsC.h, [273](#)
- PixelFormat\_SCF1WBWG10p
  - CameraDefsC.h, [273](#)
- PixelFormat\_SCF1WBWG12
  - CameraDefsC.h, [273](#)
- PixelFormat\_SCF1WBWG12p
  - CameraDefsC.h, [273](#)
- PixelFormat\_SCF1WBWG14
  - CameraDefsC.h, [273](#)
- PixelFormat\_SCF1WBWG16
  - CameraDefsC.h, [273](#)
- PixelFormat\_SCF1WBWG8
  - CameraDefsC.h, [273](#)
- PixelFormat\_SCF1WGWb10
  - CameraDefsC.h, [273](#)
- PixelFormat\_SCF1WGWb10p
  - CameraDefsC.h, [273](#)
- PixelFormat\_SCF1WGWb12
  - CameraDefsC.h, [273](#)
- PixelFormat\_SCF1WGWb12p
  - CameraDefsC.h, [273](#)
- PixelFormat\_SCF1WGWb14
  - CameraDefsC.h, [273](#)
- PixelFormat\_SCF1WGWb16
  - CameraDefsC.h, [273](#)
- PixelFormat\_SCF1WGWb8
  - CameraDefsC.h, [273](#)
- PixelFormat\_SCF1WGWb10
  - CameraDefsC.h, [273](#)
- PixelFormat\_SCF1WGWb10p
  - CameraDefsC.h, [273](#)

PixelFormat\_SCF1WGWR12  
CameraDefsC.h, [274](#)

PixelFormat\_SCF1WGWR12p  
CameraDefsC.h, [274](#)

PixelFormat\_SCF1WGWR14  
CameraDefsC.h, [274](#)

PixelFormat\_SCF1WGWR16  
CameraDefsC.h, [274](#)

PixelFormat\_SCF1WGWR8  
CameraDefsC.h, [273](#)

PixelFormat\_SCF1WRWG10  
CameraDefsC.h, [274](#)

PixelFormat\_SCF1WRWG10p  
CameraDefsC.h, [274](#)

PixelFormat\_SCF1WRWG12  
CameraDefsC.h, [274](#)

PixelFormat\_SCF1WRWG12p  
CameraDefsC.h, [274](#)

PixelFormat\_SCF1WRWG14  
CameraDefsC.h, [274](#)

PixelFormat\_SCF1WRWG16  
CameraDefsC.h, [274](#)

PixelFormat\_SCF1WRWG8  
CameraDefsC.h, [274](#)

PixelFormat\_YCbCr10\_CbYCr  
CameraDefsC.h, [274](#)

PixelFormat\_YCbCr10p\_CbYCr  
CameraDefsC.h, [274](#)

PixelFormat\_YCbCr12\_CbYCr  
CameraDefsC.h, [274](#)

PixelFormat\_YCbCr12p\_CbYCr  
CameraDefsC.h, [274](#)

PixelFormat\_YCbCr411\_8  
CameraDefsC.h, [270](#)

PixelFormat\_YCbCr411\_8\_CbYYCrYY  
CameraDefsC.h, [274](#)

PixelFormat\_YCbCr422\_10  
CameraDefsC.h, [274](#)

PixelFormat\_YCbCr422\_10\_CbYCrY  
CameraDefsC.h, [274](#)

PixelFormat\_YCbCr422\_10p  
CameraDefsC.h, [274](#)

PixelFormat\_YCbCr422\_10p\_CbYCrY  
CameraDefsC.h, [274](#)

PixelFormat\_YCbCr422\_12  
CameraDefsC.h, [274](#)

PixelFormat\_YCbCr422\_12\_CbYCrY  
CameraDefsC.h, [274](#)

PixelFormat\_YCbCr422\_12p  
CameraDefsC.h, [274](#)

PixelFormat\_YCbCr422\_12p\_CbYCrY  
CameraDefsC.h, [274](#)

PixelFormat\_YCbCr422\_8  
CameraDefsC.h, [270](#)

PixelFormat\_YCbCr422\_8\_CbYCrY  
CameraDefsC.h, [274](#)

PixelFormat\_YCbCr601\_10\_CbYCr  
CameraDefsC.h, [274](#)

PixelFormat\_YCbCr601\_12\_CbYCr  
CameraDefsC.h, [274](#)

PixelFormat\_YCbCr601\_12p\_CbYCr  
CameraDefsC.h, [274](#)

PixelFormat\_YCbCr601\_411\_8\_CbYYCrYY  
CameraDefsC.h, [274](#)

PixelFormat\_YCbCr601\_422\_10  
CameraDefsC.h, [274](#)

PixelFormat\_YCbCr601\_422\_10\_CbYCrY  
CameraDefsC.h, [274](#)

PixelFormat\_YCbCr601\_422\_10p  
CameraDefsC.h, [274](#)

PixelFormat\_YCbCr601\_422\_10p\_CbYCrY  
CameraDefsC.h, [274](#)

PixelFormat\_YCbCr601\_422\_12  
CameraDefsC.h, [274](#)

PixelFormat\_YCbCr601\_422\_12\_CbYCrY  
CameraDefsC.h, [274](#)

PixelFormat\_YCbCr601\_422\_12p  
CameraDefsC.h, [274](#)

PixelFormat\_YCbCr601\_422\_12p\_CbYCrY  
CameraDefsC.h, [274](#)

PixelFormat\_YCbCr601\_422\_8  
CameraDefsC.h, [274](#)

PixelFormat\_YCbCr601\_422\_8\_CbYCrY  
CameraDefsC.h, [274](#)

PixelFormat\_YCbCr601\_8\_CbYCr  
CameraDefsC.h, [274](#)

PixelFormat\_YCbCr709\_10\_CbYCr  
CameraDefsC.h, [274](#)

PixelFormat\_YCbCr709\_10p\_CbYCr  
CameraDefsC.h, [274](#)

PixelFormat\_YCbCr709\_12\_CbYCr  
CameraDefsC.h, [274](#)

PixelFormat\_YCbCr709\_12p\_CbYCr  
CameraDefsC.h, [274](#)

PixelFormat\_YCbCr709\_411\_8\_CbYYCrYY  
CameraDefsC.h, [275](#)

PixelFormat\_YCbCr709\_422\_10  
CameraDefsC.h, [275](#)

PixelFormat\_YCbCr709\_422\_10\_CbYCrY  
CameraDefsC.h, [275](#)

PixelFormat\_YCbCr709\_422\_10p  
CameraDefsC.h, [275](#)

PixelFormat\_YCbCr709\_422\_10p\_CbYCrY  
CameraDefsC.h, [275](#)

PixelFormat\_YCbCr709\_422\_12  
CameraDefsC.h, [275](#)

PixelFormat\_YCbCr709\_422\_12\_CbYCrY  
CameraDefsC.h, [275](#)

PixelFormat\_YCbCr709\_422\_12p  
CameraDefsC.h, [275](#)

PixelFormat\_YCbCr709\_422\_12p\_CbYCrY  
CameraDefsC.h, [275](#)

PixelFormat\_YCbCr709\_422\_8  
CameraDefsC.h, [275](#)



PixelFormat\_YCbCr709\_422\_8\_CbYCrY  
CameraDefsC.h, [275](#)

PixelFormat\_YCbCr709\_8\_CbYCr  
CameraDefsC.h, [274](#)

PixelFormat\_YCbCr8  
CameraDefsC.h, [270](#)

PixelFormat\_YCbCr8\_CbYCr  
CameraDefsC.h, [274](#)

PixelFormat\_YUV411\_8\_UYYVYY  
CameraDefsC.h, [275](#)

PixelFormat\_YUV411Packed  
CameraDefsC.h, [270](#)

PixelFormat\_YUV422\_8  
CameraDefsC.h, [275](#)

PixelFormat\_YUV422\_8\_UYVY  
CameraDefsC.h, [275](#)

PixelFormat\_YUV422Packed  
CameraDefsC.h, [270](#)

PixelFormat\_YUV444Packed  
CameraDefsC.h, [270](#)

PixelFormat\_YUV8\_UYV  
CameraDefsC.h, [275](#)

PixelFormatInfoID  
\_quickSpin, [131](#)

PixelFormatInfoSelector  
\_quickSpin, [131](#)

PixelFormatInfoSelector\_B10  
CameraDefsC.h, [277](#)

PixelFormatInfoSelector\_B12  
CameraDefsC.h, [277](#)

PixelFormatInfoSelector\_B16  
CameraDefsC.h, [277](#)

PixelFormatInfoSelector\_B8  
CameraDefsC.h, [277](#)

PixelFormatInfoSelector\_BayerBG10  
CameraDefsC.h, [276](#)

PixelFormatInfoSelector\_BayerBG10p  
CameraDefsC.h, [276](#)

PixelFormatInfoSelector\_BayerBG12  
CameraDefsC.h, [276](#)

PixelFormatInfoSelector\_BayerBG12p  
CameraDefsC.h, [276](#)

PixelFormatInfoSelector\_BayerBG16  
CameraDefsC.h, [276](#)

PixelFormatInfoSelector\_BayerBG8  
CameraDefsC.h, [276](#)

PixelFormatInfoSelector\_BayerGB10  
CameraDefsC.h, [276](#)

PixelFormatInfoSelector\_BayerGB10p  
CameraDefsC.h, [276](#)

PixelFormatInfoSelector\_BayerGB12  
CameraDefsC.h, [276](#)

PixelFormatInfoSelector\_BayerGB12p  
CameraDefsC.h, [276](#)

PixelFormatInfoSelector\_BayerGB16  
CameraDefsC.h, [276](#)

PixelFormatInfoSelector\_BayerGB8  
CameraDefsC.h, [276](#)

PixelFormatInfoSelector\_BayerGR10  
CameraDefsC.h, [276](#)

PixelFormatInfoSelector\_BayerGR10p  
CameraDefsC.h, [276](#)

PixelFormatInfoSelector\_BayerGR12  
CameraDefsC.h, [276](#)

PixelFormatInfoSelector\_BayerGR12p  
CameraDefsC.h, [276](#)

PixelFormatInfoSelector\_BayerGR16  
CameraDefsC.h, [276](#)

PixelFormatInfoSelector\_BayerGR8  
CameraDefsC.h, [276](#)

PixelFormatInfoSelector\_BayerRG10  
CameraDefsC.h, [276](#)

PixelFormatInfoSelector\_BayerRG10p  
CameraDefsC.h, [276](#)

PixelFormatInfoSelector\_BayerRG12  
CameraDefsC.h, [276](#)

PixelFormatInfoSelector\_BayerRG12p  
CameraDefsC.h, [276](#)

PixelFormatInfoSelector\_BayerRG16  
CameraDefsC.h, [276](#)

PixelFormatInfoSelector\_BayerRG8  
CameraDefsC.h, [276](#)

PixelFormatInfoSelector\_BayerRGPolarized10p  
CameraDefsC.h, [281](#)

PixelFormatInfoSelector\_BayerRGPolarized12p  
CameraDefsC.h, [281](#)

PixelFormatInfoSelector\_BayerRGPolarized16  
CameraDefsC.h, [281](#)

PixelFormatInfoSelector\_BayerRGPolarized8  
CameraDefsC.h, [281](#)

PixelFormatInfoSelector\_BGR10  
CameraDefsC.h, [277](#)

PixelFormatInfoSelector\_BGR10p  
CameraDefsC.h, [277](#)

PixelFormatInfoSelector\_BGR12  
CameraDefsC.h, [277](#)

PixelFormatInfoSelector\_BGR12p  
CameraDefsC.h, [277](#)

PixelFormatInfoSelector\_BGR14  
CameraDefsC.h, [277](#)

PixelFormatInfoSelector\_BGR16  
CameraDefsC.h, [277](#)

PixelFormatInfoSelector\_BGR565p  
CameraDefsC.h, [277](#)

PixelFormatInfoSelector\_BGR8  
CameraDefsC.h, [277](#)

PixelFormatInfoSelector\_BGRa10  
CameraDefsC.h, [277](#)

PixelFormatInfoSelector\_BGRa10p  
CameraDefsC.h, [277](#)

PixelFormatInfoSelector\_BGRa12  
CameraDefsC.h, [277](#)

PixelFormatInfoSelector\_BGRa12p  
CameraDefsC.h, [277](#)

PixelFormatInfoSelector\_BGRa14  
CameraDefsC.h, [277](#)

- PixelFormatInfoSelector\_BGRa16  
CameraDefsC.h, [277](#)
- PixelFormatInfoSelector\_BGRa8  
CameraDefsC.h, [277](#)
- PixelFormatInfoSelector\_BiColorBGRG10  
CameraDefsC.h, [278](#)
- PixelFormatInfoSelector\_BiColorBGRG10p  
CameraDefsC.h, [278](#)
- PixelFormatInfoSelector\_BiColorBGRG12  
CameraDefsC.h, [278](#)
- PixelFormatInfoSelector\_BiColorBGRG12p  
CameraDefsC.h, [278](#)
- PixelFormatInfoSelector\_BiColorBGRG8  
CameraDefsC.h, [278](#)
- PixelFormatInfoSelector\_BiColorRGBG10  
CameraDefsC.h, [278](#)
- PixelFormatInfoSelector\_BiColorRGBG10p  
CameraDefsC.h, [278](#)
- PixelFormatInfoSelector\_BiColorRGBG12  
CameraDefsC.h, [278](#)
- PixelFormatInfoSelector\_BiColorRGBG12p  
CameraDefsC.h, [278](#)
- PixelFormatInfoSelector\_BiColorRGBG8  
CameraDefsC.h, [278](#)
- PixelFormatInfoSelector\_Confidence1  
CameraDefsC.h, [278](#)
- PixelFormatInfoSelector\_Confidence16  
CameraDefsC.h, [278](#)
- PixelFormatInfoSelector\_Confidence1p  
CameraDefsC.h, [278](#)
- PixelFormatInfoSelector\_Confidence32f  
CameraDefsC.h, [278](#)
- PixelFormatInfoSelector\_Confidence8  
CameraDefsC.h, [278](#)
- PixelFormatInfoSelector\_Coord3D\_A10p  
CameraDefsC.h, [278](#)
- PixelFormatInfoSelector\_Coord3D\_A12p  
CameraDefsC.h, [278](#)
- PixelFormatInfoSelector\_Coord3D\_A16  
CameraDefsC.h, [278](#)
- PixelFormatInfoSelector\_Coord3D\_A32f  
CameraDefsC.h, [278](#)
- PixelFormatInfoSelector\_Coord3D\_A8  
CameraDefsC.h, [278](#)
- PixelFormatInfoSelector\_Coord3D\_ABC10p  
CameraDefsC.h, [277](#)
- PixelFormatInfoSelector\_Coord3D\_ABC10p\_Planar  
CameraDefsC.h, [277](#)
- PixelFormatInfoSelector\_Coord3D\_ABC12p  
CameraDefsC.h, [277](#)
- PixelFormatInfoSelector\_Coord3D\_ABC12p\_Planar  
CameraDefsC.h, [277](#)
- PixelFormatInfoSelector\_Coord3D\_ABC16  
CameraDefsC.h, [277](#)
- PixelFormatInfoSelector\_Coord3D\_ABC16\_Planar  
CameraDefsC.h, [278](#)
- PixelFormatInfoSelector\_Coord3D\_ABC32f  
CameraDefsC.h, [278](#)
- PixelFormatInfoSelector\_Coord3D\_ABC32f\_Planar  
CameraDefsC.h, [278](#)
- PixelFormatInfoSelector\_Coord3D\_ABC8  
CameraDefsC.h, [278](#)
- PixelFormatInfoSelector\_Coord3D\_ABC8\_Planar  
CameraDefsC.h, [278](#)
- PixelFormatInfoSelector\_Coord3D\_B10p  
CameraDefsC.h, [278](#)
- PixelFormatInfoSelector\_Coord3D\_B12p  
CameraDefsC.h, [278](#)
- PixelFormatInfoSelector\_Coord3D\_B16  
CameraDefsC.h, [278](#)
- PixelFormatInfoSelector\_Coord3D\_B32f  
CameraDefsC.h, [278](#)
- PixelFormatInfoSelector\_Coord3D\_B8  
CameraDefsC.h, [278](#)
- PixelFormatInfoSelector\_Coord3D\_C10p  
CameraDefsC.h, [278](#)
- PixelFormatInfoSelector\_Coord3D\_C12p  
CameraDefsC.h, [278](#)
- PixelFormatInfoSelector\_Coord3D\_C16  
CameraDefsC.h, [278](#)
- PixelFormatInfoSelector\_Coord3D\_C32f  
CameraDefsC.h, [278](#)
- PixelFormatInfoSelector\_Coord3D\_C8  
CameraDefsC.h, [278](#)
- PixelFormatInfoSelector\_G10  
CameraDefsC.h, [277](#)
- PixelFormatInfoSelector\_G12  
CameraDefsC.h, [277](#)
- PixelFormatInfoSelector\_G16  
CameraDefsC.h, [277](#)
- PixelFormatInfoSelector\_G8  
CameraDefsC.h, [277](#)
- PixelFormatInfoSelector\_JPEGColor8  
CameraDefsC.h, [281](#)
- PixelFormatInfoSelector\_JPEGMono8  
CameraDefsC.h, [281](#)



- PixelFormatInfoSelector\_LLCBayerRG8  
CameraDefsC.h, [281](#)
- PixelFormatInfoSelector\_LLCMono8  
CameraDefsC.h, [281](#)
- PixelFormatInfoSelector\_Mono10  
CameraDefsC.h, [276](#)
- PixelFormatInfoSelector\_Mono10p  
CameraDefsC.h, [276](#)
- PixelFormatInfoSelector\_Mono12  
CameraDefsC.h, [276](#)
- PixelFormatInfoSelector\_Mono12p  
CameraDefsC.h, [276](#)
- PixelFormatInfoSelector\_Mono14  
CameraDefsC.h, [276](#)
- PixelFormatInfoSelector\_Mono16  
CameraDefsC.h, [276](#)
- PixelFormatInfoSelector\_Mono16s  
CameraDefsC.h, [276](#)
- PixelFormatInfoSelector\_Mono1p  
CameraDefsC.h, [275](#)
- PixelFormatInfoSelector\_Mono2p  
CameraDefsC.h, [276](#)
- PixelFormatInfoSelector\_Mono32f  
CameraDefsC.h, [276](#)
- PixelFormatInfoSelector\_Mono4p  
CameraDefsC.h, [276](#)
- PixelFormatInfoSelector\_Mono8  
CameraDefsC.h, [276](#)
- PixelFormatInfoSelector\_Mono8s  
CameraDefsC.h, [276](#)
- PixelFormatInfoSelector\_Polarized10p  
CameraDefsC.h, [281](#)
- PixelFormatInfoSelector\_Polarized12p  
CameraDefsC.h, [281](#)
- PixelFormatInfoSelector\_Polarized16  
CameraDefsC.h, [281](#)
- PixelFormatInfoSelector\_Polarized8  
CameraDefsC.h, [281](#)
- PixelFormatInfoSelector\_R10  
CameraDefsC.h, [277](#)
- PixelFormatInfoSelector\_R12  
CameraDefsC.h, [277](#)
- PixelFormatInfoSelector\_R16  
CameraDefsC.h, [277](#)
- PixelFormatInfoSelector\_R8  
CameraDefsC.h, [277](#)
- PixelFormatInfoSelector\_RGB10  
CameraDefsC.h, [276](#)
- PixelFormatInfoSelector\_RGB10\_Planar  
CameraDefsC.h, [276](#)
- PixelFormatInfoSelector\_RGB10p  
CameraDefsC.h, [277](#)
- PixelFormatInfoSelector\_RGB10p32  
CameraDefsC.h, [277](#)
- PixelFormatInfoSelector\_RGB12  
CameraDefsC.h, [277](#)
- PixelFormatInfoSelector\_RGB12\_Planar  
CameraDefsC.h, [277](#)
- PixelFormatInfoSelector\_RGB12p  
CameraDefsC.h, [277](#)
- PixelFormatInfoSelector\_RGB14  
CameraDefsC.h, [277](#)
- PixelFormatInfoSelector\_RGB16  
CameraDefsC.h, [277](#)
- PixelFormatInfoSelector\_RGB16\_Planar  
CameraDefsC.h, [277](#)
- PixelFormatInfoSelector\_RGB16s  
CameraDefsC.h, [277](#)
- PixelFormatInfoSelector\_RGB32f  
CameraDefsC.h, [277](#)
- PixelFormatInfoSelector\_RGB565p  
CameraDefsC.h, [277](#)
- PixelFormatInfoSelector\_RGB8  
CameraDefsC.h, [276](#)
- PixelFormatInfoSelector\_RGB8\_Planar  
CameraDefsC.h, [276](#)
- PixelFormatInfoSelector\_RGBa10  
CameraDefsC.h, [276](#)
- PixelFormatInfoSelector\_RGBa10p  
CameraDefsC.h, [276](#)
- PixelFormatInfoSelector\_RGBa12  
CameraDefsC.h, [276](#)
- PixelFormatInfoSelector\_RGBa12p  
CameraDefsC.h, [276](#)
- PixelFormatInfoSelector\_RGBa14  
CameraDefsC.h, [276](#)
- PixelFormatInfoSelector\_RGBa16  
CameraDefsC.h, [276](#)
- PixelFormatInfoSelector\_RGBa32f  
CameraDefsC.h, [277](#)
- PixelFormatInfoSelector\_RGBa8  
CameraDefsC.h, [276](#)
- PixelFormatInfoSelector\_SCF1WBWG10  
CameraDefsC.h, [278](#)
- PixelFormatInfoSelector\_SCF1WBWG10p  
CameraDefsC.h, [279](#)
- PixelFormatInfoSelector\_SCF1WBWG12  
CameraDefsC.h, [279](#)
- PixelFormatInfoSelector\_SCF1WBWG12p  
CameraDefsC.h, [279](#)
- PixelFormatInfoSelector\_SCF1WBWG14  
CameraDefsC.h, [279](#)
- PixelFormatInfoSelector\_SCF1WBWG16  
CameraDefsC.h, [279](#)
- PixelFormatInfoSelector\_SCF1WBWG8  
CameraDefsC.h, [278](#)
- PixelFormatInfoSelector\_SCF1WGWB10  
CameraDefsC.h, [279](#)
- PixelFormatInfoSelector\_SCF1WGWB10p  
CameraDefsC.h, [279](#)
- PixelFormatInfoSelector\_SCF1WGWB12  
CameraDefsC.h, [279](#)
- PixelFormatInfoSelector\_SCF1WGWB12p  
CameraDefsC.h, [279](#)
- PixelFormatInfoSelector\_SCF1WGWB14  
CameraDefsC.h, [279](#)

- PixelFormatInfoSelector\_SCF1WGWB16  
CameraDefsC.h, [279](#)
- PixelFormatInfoSelector\_SCF1WGWB8  
CameraDefsC.h, [279](#)
- PixelFormatInfoSelector\_SCF1WGWR10  
CameraDefsC.h, [279](#)
- PixelFormatInfoSelector\_SCF1WGWR10p  
CameraDefsC.h, [279](#)
- PixelFormatInfoSelector\_SCF1WGWR12  
CameraDefsC.h, [279](#)
- PixelFormatInfoSelector\_SCF1WGWR12p  
CameraDefsC.h, [279](#)
- PixelFormatInfoSelector\_SCF1WGWR14  
CameraDefsC.h, [279](#)
- PixelFormatInfoSelector\_SCF1WGWR16  
CameraDefsC.h, [279](#)
- PixelFormatInfoSelector\_SCF1WGWR8  
CameraDefsC.h, [279](#)
- PixelFormatInfoSelector\_SCF1WRWG10  
CameraDefsC.h, [279](#)
- PixelFormatInfoSelector\_SCF1WRWG10p  
CameraDefsC.h, [279](#)
- PixelFormatInfoSelector\_SCF1WRWG12  
CameraDefsC.h, [279](#)
- PixelFormatInfoSelector\_SCF1WRWG12p  
CameraDefsC.h, [279](#)
- PixelFormatInfoSelector\_SCF1WRWG14  
CameraDefsC.h, [279](#)
- PixelFormatInfoSelector\_SCF1WRWG16  
CameraDefsC.h, [279](#)
- PixelFormatInfoSelector\_SCF1WRWG8  
CameraDefsC.h, [279](#)
- PixelFormatInfoSelector\_YCbCr10\_CbYCr  
CameraDefsC.h, [279](#)
- PixelFormatInfoSelector\_YCbCr10p\_CbYCr  
CameraDefsC.h, [279](#)
- PixelFormatInfoSelector\_YCbCr12\_CbYCr  
CameraDefsC.h, [280](#)
- PixelFormatInfoSelector\_YCbCr12p\_CbYCr  
CameraDefsC.h, [280](#)
- PixelFormatInfoSelector\_YCbCr411\_8  
CameraDefsC.h, [280](#)
- PixelFormatInfoSelector\_YCbCr411\_8\_CbYYCrYY  
CameraDefsC.h, [280](#)
- PixelFormatInfoSelector\_YCbCr422\_10  
CameraDefsC.h, [280](#)
- PixelFormatInfoSelector\_YCbCr422\_10\_CbYCrY  
CameraDefsC.h, [280](#)
- PixelFormatInfoSelector\_YCbCr422\_10p  
CameraDefsC.h, [280](#)
- PixelFormatInfoSelector\_YCbCr422\_10p\_CbYCrY  
CameraDefsC.h, [280](#)
- PixelFormatInfoSelector\_YCbCr422\_12  
CameraDefsC.h, [280](#)
- PixelFormatInfoSelector\_YCbCr422\_12\_CbYCrY  
CameraDefsC.h, [280](#)
- PixelFormatInfoSelector\_YCbCr422\_12p  
CameraDefsC.h, [280](#)
- PixelFormatInfoSelector\_YCbCr422\_12p\_CbYCrY  
CameraDefsC.h, [280](#)
- PixelFormatInfoSelector\_YCbCr422\_8  
CameraDefsC.h, [280](#)
- PixelFormatInfoSelector\_YCbCr422\_8\_CbYCrY  
CameraDefsC.h, [280](#)
- PixelFormatInfoSelector\_YCbCr601\_8\_CbYCr  
CameraDefsC.h, [280](#)
- PixelFormatInfoSelector\_YCbCr709\_10\_CbYCr  
CameraDefsC.h, [280](#)
- PixelFormatInfoSelector\_YCbCr709\_10p\_CbYCr  
CameraDefsC.h, [280](#)
- PixelFormatInfoSelector\_YCbCr709\_12\_CbYCr  
CameraDefsC.h, [280](#)
- PixelFormatInfoSelector\_YCbCr709\_12p\_CbYCr  
CameraDefsC.h, [280](#)
- PixelFormatInfoSelector\_YCbCr709\_411\_8\_CbYYCrYY  
CameraDefsC.h, [280](#)
- PixelFormatInfoSelector\_YCbCr709\_422\_10  
CameraDefsC.h, [280](#)
- PixelFormatInfoSelector\_YCbCr709\_422\_10\_CbYCrY  
CameraDefsC.h, [280](#)
- PixelFormatInfoSelector\_YCbCr709\_422\_10p  
CameraDefsC.h, [280](#)
- PixelFormatInfoSelector\_YCbCr709\_422\_10p\_CbYCrY  
CameraDefsC.h, [281](#)
- PixelFormatInfoSelector\_YCbCr709\_422\_12  
CameraDefsC.h, [281](#)

- PixelFormatInfoSelector\_YCbCr709\_422\_12\_CbYCrY
  - CameraDefsC.h, [281](#)
- PixelFormatInfoSelector\_YCbCr709\_422\_12p
  - CameraDefsC.h, [281](#)
- PixelFormatInfoSelector\_YCbCr709\_422\_12p\_CbYCrY
  - CameraDefsC.h, [281](#)
- PixelFormatInfoSelector\_YCbCr709\_422\_8
  - CameraDefsC.h, [280](#)
- PixelFormatInfoSelector\_YCbCr709\_422\_8\_CbYCrY
  - CameraDefsC.h, [280](#)
- PixelFormatInfoSelector\_YCbCr709\_8\_CbYCr
  - CameraDefsC.h, [280](#)
- PixelFormatInfoSelector\_YCbCr8
  - CameraDefsC.h, [279](#)
- PixelFormatInfoSelector\_YCbCr8\_CbYCr
  - CameraDefsC.h, [279](#)
- PixelFormatInfoSelector\_YUV411\_8\_UYYVYY
  - CameraDefsC.h, [281](#)
- PixelFormatInfoSelector\_YUV422\_8
  - CameraDefsC.h, [281](#)
- PixelFormatInfoSelector\_YUV422\_8\_UYVY
  - CameraDefsC.h, [281](#)
- PixelFormatInfoSelector\_YUV8\_UYV
  - CameraDefsC.h, [281](#)
- PixelSize
  - \_quickSpin, [131](#)
- PixelSize\_Bpp1
  - CameraDefsC.h, [281](#)
- PixelSize\_Bpp10
  - CameraDefsC.h, [281](#)
- PixelSize\_Bpp12
  - CameraDefsC.h, [281](#)
- PixelSize\_Bpp14
  - CameraDefsC.h, [281](#)
- PixelSize\_Bpp16
  - CameraDefsC.h, [281](#)
- PixelSize\_Bpp2
  - CameraDefsC.h, [281](#)
- PixelSize\_Bpp20
  - CameraDefsC.h, [281](#)
- PixelSize\_Bpp24
  - CameraDefsC.h, [281](#)
- PixelSize\_Bpp30
  - CameraDefsC.h, [281](#)
- PixelSize\_Bpp32
  - CameraDefsC.h, [281](#)
- PixelSize\_Bpp36
  - CameraDefsC.h, [282](#)
- PixelSize\_Bpp4
  - CameraDefsC.h, [281](#)
- PixelSize\_Bpp48
  - CameraDefsC.h, [282](#)
- PixelSize\_Bpp64
  - CameraDefsC.h, [282](#)
- PixelSize\_Bpp8
  - CameraDefsC.h, [281](#)
- PixelSize\_Bpp96
  - CameraDefsC.h, [282](#)
- PNG
  - SpinnakerDefsC.h, [411](#)
- POEStatus
  - \_quickSpinTLInterface, [161](#)
- POEStatus\_NotSupported
  - TransportLayerDefsC.h, [473](#)
- POEStatus\_PowerOff
  - TransportLayerDefsC.h, [473](#)
- POEStatus\_PowerOn
  - TransportLayerDefsC.h, [473](#)
- PortNode
  - SpinnakerGenApiDefsC.h, [461](#)
- PowerSupplyCurrent
  - \_quickSpin, [132](#)
- PowerSupplyVoltage
  - \_quickSpin, [132](#)
- PPM
  - SpinnakerDefsC.h, [410](#)
- progressive
  - \_spinJPEGOption, [179](#)
- PureNumber
  - SpinnakerGenApiDefsC.h, [461](#)
- quality
  - \_spinJPEGOption, [179](#)
  - \_spinJPG2Option, [180](#)
  - \_spinMJPGOption, [182](#)
- QuickSpin Access, [11](#)
- quickSpinBooleanNode
  - QuickSpinDefsC.h, [306](#)
- QuickSpinC.h
  - quickSpinInit, [304](#)
  - quickSpinInitEx, [304](#)
  - quickSpinTLDeviceInit, [305](#)
  - quickSpinTLInterfaceInit, [305](#)
  - quickSpinTLStreamInit, [305](#)
  - quickSpinTLSystemInit, [305](#)
- quickSpinCommandNode
  - QuickSpinDefsC.h, [307](#)
- QuickSpinDefsC.h
  - quickSpinBooleanNode, [306](#)
  - quickSpinCommandNode, [307](#)
  - quickSpinEnumerationNode, [307](#)
  - quickSpinFloatNode, [307](#)
  - quickSpinIntegerNode, [307](#)
  - quickSpinRegisterNode, [307](#)
  - quickSpinStringNode, [307](#)
- quickSpinEnumerationNode
  - QuickSpinDefsC.h, [307](#)
- quickSpinFloatNode
  - QuickSpinDefsC.h, [307](#)
- quickSpinInit
  - QuickSpinC.h, [304](#)
- quickSpinInitEx
  - QuickSpinC.h, [304](#)
- quickSpinIntegerNode
  - QuickSpinDefsC.h, [307](#)
- quickSpinRegisterNode
  - QuickSpinDefsC.h, [307](#)

- quickSpinStringNode
  - QuickSpinDefsC.h, [307](#)
- quickSpinTLDeviceInit
  - QuickSpinC.h, [305](#)
- quickSpinTLInterfaceInit
  - QuickSpinC.h, [305](#)
- quickSpinTLStreamInit
  - QuickSpinC.h, [305](#)
- quickSpinTLSystemInit
  - QuickSpinC.h, [305](#)
- RAW
  - SpinnakerDefsC.h, [411](#)
- RED
  - SpinnakerDefsC.h, [413](#)
- RegionDestination
  - \_quickSpin, [132](#)
- RegionDestination\_Stream0
  - CameraDefsC.h, [282](#)
- RegionDestination\_Stream1
  - CameraDefsC.h, [282](#)
- RegionDestination\_Stream2
  - CameraDefsC.h, [282](#)
- RegionMode
  - \_quickSpin, [132](#)
- RegionMode\_Off
  - CameraDefsC.h, [282](#)
- RegionMode\_On
  - CameraDefsC.h, [282](#)
- RegionSelector
  - \_quickSpin, [132](#)
- RegionSelector\_All
  - CameraDefsC.h, [283](#)
- RegionSelector\_Region0
  - CameraDefsC.h, [283](#)
- RegionSelector\_Region1
  - CameraDefsC.h, [283](#)
- RegionSelector\_Region2
  - CameraDefsC.h, [283](#)
- RegisterNode
  - SpinnakerGenApiDefsC.h, [461](#)
- reserved
  - \_spinAVIOption, [170](#)
  - \_spinBMPOption, [171](#)
  - \_spinH264Option, [178](#)
  - \_spinJPEGOption, [179](#)
  - \_spinJPG2Option, [180](#)
  - \_spinMJPEGOption, [182](#)
  - \_spinPGMOption, [183](#)
  - \_spinPNGOption, [184](#)
  - \_spinPPMOption, [185](#)
  - \_spinTIFFOption, [186](#)
- ReverseX
  - \_quickSpin, [132](#)
- ReverseY
  - \_quickSpin, [132](#)
- RgbTransformLightSource
  - \_quickSpin, [132](#)
- RgbTransformLightSource\_Cloudy6500K
  - CameraDefsC.h, [283](#)
- RgbTransformLightSource\_CoolFluorescent4000K
  - CameraDefsC.h, [283](#)
- RgbTransformLightSource\_Custom
  - CameraDefsC.h, [283](#)
- RgbTransformLightSource\_Daylight5000K
  - CameraDefsC.h, [283](#)
- RgbTransformLightSource\_General
  - CameraDefsC.h, [283](#)
- RgbTransformLightSource\_Shade8000K
  - CameraDefsC.h, [283](#)
- RgbTransformLightSource\_Tungsten2800K
  - CameraDefsC.h, [283](#)
- RgbTransformLightSource\_WarmFluorescent3000K
  - CameraDefsC.h, [283](#)
- RIGOROUS
  - SpinnakerDefsC.h, [409](#)
- RO
  - SpinnakerGenApiDefsC.h, [457](#)
- RW
  - SpinnakerGenApiDefsC.h, [457](#)
- SATURATION
  - SpinnakerDefsC.h, [413](#)
- Saturation
  - \_quickSpin, [133](#)
- SaturationEnable
  - \_quickSpin, [133](#)
- Scan3dAxisMax
  - \_quickSpin, [133](#)
- Scan3dAxisMin
  - \_quickSpin, [133](#)
- Scan3dCoordinateOffset
  - \_quickSpin, [133](#)
- Scan3dCoordinateReferenceSelector
  - \_quickSpin, [133](#)
- Scan3dCoordinateReferenceSelector\_RotationX
  - CameraDefsC.h, [284](#)
- Scan3dCoordinateReferenceSelector\_RotationY
  - CameraDefsC.h, [284](#)
- Scan3dCoordinateReferenceSelector\_RotationZ
  - CameraDefsC.h, [284](#)
- Scan3dCoordinateReferenceSelector\_TranslationX
  - CameraDefsC.h, [284](#)
- Scan3dCoordinateReferenceSelector\_TranslationY
  - CameraDefsC.h, [284](#)
- Scan3dCoordinateReferenceSelector\_TranslationZ
  - CameraDefsC.h, [284](#)
- Scan3dCoordinateReferenceValue
  - \_quickSpin, [133](#)
- Scan3dCoordinateScale
  - \_quickSpin, [133](#)
- Scan3dCoordinateSelector
  - \_quickSpin, [134](#)
- Scan3dCoordinateSelector\_CoordinateA
  - CameraDefsC.h, [284](#)
- Scan3dCoordinateSelector\_CoordinateB
  - CameraDefsC.h, [284](#)
- Scan3dCoordinateSelector\_CoordinateC
  - CameraDefsC.h, [284](#)

- CameraDefsC.h, [284](#)
- Scan3dCoordinateSystem
  - \_quickSpin, [134](#)
- Scan3dCoordinateSystem\_Cartesian
  - CameraDefsC.h, [284](#)
- Scan3dCoordinateSystem\_Cylindrical
  - CameraDefsC.h, [284](#)
- Scan3dCoordinateSystem\_Spherical
  - CameraDefsC.h, [284](#)
- Scan3dCoordinateSystemReference
  - \_quickSpin, [134](#)
- Scan3dCoordinateSystemReference\_Anchor
  - CameraDefsC.h, [285](#)
- Scan3dCoordinateSystemReference\_Transformed
  - CameraDefsC.h, [285](#)
- Scan3dCoordinateTransformSelector
  - \_quickSpin, [134](#)
- Scan3dCoordinateTransformSelector\_RotationX
  - CameraDefsC.h, [285](#)
- Scan3dCoordinateTransformSelector\_RotationY
  - CameraDefsC.h, [285](#)
- Scan3dCoordinateTransformSelector\_RotationZ
  - CameraDefsC.h, [285](#)
- Scan3dCoordinateTransformSelector\_TranslationX
  - CameraDefsC.h, [285](#)
- Scan3dCoordinateTransformSelector\_TranslationY
  - CameraDefsC.h, [285](#)
- Scan3dCoordinateTransformSelector\_TranslationZ
  - CameraDefsC.h, [285](#)
- Scan3dDistanceUnit
  - \_quickSpin, [134](#)
- Scan3dDistanceUnit\_Inch
  - CameraDefsC.h, [285](#)
- Scan3dDistanceUnit\_Millimeter
  - CameraDefsC.h, [285](#)
- Scan3dInvalidDataFlag
  - \_quickSpin, [134](#)
- Scan3dInvalidDataValue
  - \_quickSpin, [134](#)
- Scan3dOutputMode
  - \_quickSpin, [134](#)
- Scan3dOutputMode\_CalibratedABC\_Grid
  - CameraDefsC.h, [286](#)
- Scan3dOutputMode\_CalibratedABC\_PointCloud
  - CameraDefsC.h, [286](#)
- Scan3dOutputMode\_CalibratedAC
  - CameraDefsC.h, [286](#)
- Scan3dOutputMode\_CalibratedAC\_Linescan
  - CameraDefsC.h, [286](#)
- Scan3dOutputMode\_CalibratedC
  - CameraDefsC.h, [286](#)
- Scan3dOutputMode\_CalibratedC\_Linescan
  - CameraDefsC.h, [286](#)
- Scan3dOutputMode\_DisparityC
  - CameraDefsC.h, [286](#)
- Scan3dOutputMode\_DisparityC\_Linescan
  - CameraDefsC.h, [286](#)
- Scan3dOutputMode\_RectifiedC
  - CameraDefsC.h, [286](#)
- Scan3dOutputMode\_RectifiedC\_Linescan
  - CameraDefsC.h, [286](#)
- Scan3dOutputMode\_UncalibratedC
  - CameraDefsC.h, [286](#)
- Scan3dTransformValue
  - \_quickSpin, [135](#)
- SensorDescription
  - \_quickSpin, [135](#)
- SensorDigitizationTaps
  - \_quickSpin, [135](#)
- SensorDigitizationTaps\_Eight
  - CameraDefsC.h, [287](#)
- SensorDigitizationTaps\_Four
  - CameraDefsC.h, [287](#)
- SensorDigitizationTaps\_One
  - CameraDefsC.h, [286](#)
- SensorDigitizationTaps\_Ten
  - CameraDefsC.h, [287](#)
- SensorDigitizationTaps\_Three
  - CameraDefsC.h, [287](#)
- SensorDigitizationTaps\_Two
  - CameraDefsC.h, [287](#)
- SensorHeight
  - \_quickSpin, [135](#)
- SensorShutterMode
  - \_quickSpin, [135](#)
- SensorShutterMode\_Global
  - CameraDefsC.h, [287](#)
- SensorShutterMode\_GlobalReset
  - CameraDefsC.h, [287](#)
- SensorShutterMode\_Rolling
  - CameraDefsC.h, [287](#)
- SensorTaps
  - \_quickSpin, [135](#)
- SensorTaps\_Eight
  - CameraDefsC.h, [287](#)
- SensorTaps\_Four
  - CameraDefsC.h, [287](#)
- SensorTaps\_One
  - CameraDefsC.h, [287](#)
- SensorTaps\_Ten
  - CameraDefsC.h, [287](#)
- SensorTaps\_Three
  - CameraDefsC.h, [287](#)
- SensorTaps\_Two
  - CameraDefsC.h, [287](#)
- SensorWidth
  - \_quickSpin, [135](#)
- SequencerConfigurationMode
  - \_quickSpin, [135](#)
- SequencerConfigurationMode\_Off
  - CameraDefsC.h, [288](#)
- SequencerConfigurationMode\_On
  - CameraDefsC.h, [288](#)
- SequencerConfigurationValid
  - \_quickSpin, [136](#)
- SequencerConfigurationValid\_No

- CameraDefsC.h, [288](#)
- SequencerConfigurationValid\_Yes
  - CameraDefsC.h, [288](#)
- SequencerFeatureEnable
  - \_quickSpin, [136](#)
- SequencerMode
  - \_quickSpin, [136](#)
- SequencerMode\_Off
  - CameraDefsC.h, [288](#)
- SequencerMode\_On
  - CameraDefsC.h, [288](#)
- SequencerPathSelector
  - \_quickSpin, [136](#)
- SequencerSetActive
  - \_quickSpin, [136](#)
- SequencerSetLoad
  - \_quickSpin, [136](#)
- SequencerSetNext
  - \_quickSpin, [136](#)
- SequencerSetSave
  - \_quickSpin, [136](#)
- SequencerSetSelector
  - \_quickSpin, [137](#)
- SequencerSetStart
  - \_quickSpin, [137](#)
- SequencerSetValid
  - \_quickSpin, [137](#)
- SequencerSetValid\_No
  - CameraDefsC.h, [289](#)
- SequencerSetValid\_Yes
  - CameraDefsC.h, [289](#)
- SequencerTriggerActivation
  - \_quickSpin, [137](#)
- SequencerTriggerActivation\_AnyEdge
  - CameraDefsC.h, [289](#)
- SequencerTriggerActivation\_FallingEdge
  - CameraDefsC.h, [289](#)
- SequencerTriggerActivation\_LevelHigh
  - CameraDefsC.h, [289](#)
- SequencerTriggerActivation\_LevelLow
  - CameraDefsC.h, [289](#)
- SequencerTriggerActivation\_RisingEdge
  - CameraDefsC.h, [289](#)
- SequencerTriggerSource
  - \_quickSpin, [137](#)
- SequencerTriggerSource\_FrameStart
  - CameraDefsC.h, [289](#)
- SequencerTriggerSource\_Off
  - CameraDefsC.h, [289](#)
- SerialPortBaudRate
  - \_quickSpin, [137](#)
- SerialPortBaudRate\_Baud115200
  - CameraDefsC.h, [290](#)
- SerialPortBaudRate\_Baud1200
  - CameraDefsC.h, [290](#)
- SerialPortBaudRate\_Baud14400
  - CameraDefsC.h, [290](#)
- SerialPortBaudRate\_Baud19200
  - CameraDefsC.h, [290](#)
- SerialPortBaudRate\_Baud230400
  - CameraDefsC.h, [290](#)
- SerialPortBaudRate\_Baud2400
  - CameraDefsC.h, [290](#)
- SerialPortBaudRate\_Baud300
  - CameraDefsC.h, [290](#)
- SerialPortBaudRate\_Baud38400
  - CameraDefsC.h, [290](#)
- SerialPortBaudRate\_Baud460800
  - CameraDefsC.h, [290](#)
- SerialPortBaudRate\_Baud4800
  - CameraDefsC.h, [290](#)
- SerialPortBaudRate\_Baud57600
  - CameraDefsC.h, [290](#)
- SerialPortBaudRate\_Baud600
  - CameraDefsC.h, [290](#)
- SerialPortBaudRate\_Baud921600
  - CameraDefsC.h, [290](#)
- SerialPortBaudRate\_Baud9600
  - CameraDefsC.h, [290](#)
- SerialPortDataBits
  - \_quickSpin, [137](#)
- SerialPortParity
  - \_quickSpin, [137](#)
- SerialPortParity\_Even
  - CameraDefsC.h, [290](#)
- SerialPortParity\_Mark
  - CameraDefsC.h, [290](#)
- SerialPortParity\_None
  - CameraDefsC.h, [290](#)
- SerialPortParity\_Odd
  - CameraDefsC.h, [290](#)
- SerialPortParity\_Space
  - CameraDefsC.h, [290](#)
- SerialPortSelector
  - \_quickSpin, [138](#)
- SerialPortSelector\_SerialPort0
  - CameraDefsC.h, [290](#)
- SerialPortSource
  - \_quickSpin, [138](#)
- SerialPortSource\_Line0
  - CameraDefsC.h, [291](#)
- SerialPortSource\_Line1
  - CameraDefsC.h, [291](#)
- SerialPortSource\_Line2
  - CameraDefsC.h, [291](#)
- SerialPortSource\_Line3
  - CameraDefsC.h, [291](#)
- SerialPortSource\_Off
  - CameraDefsC.h, [291](#)
- SerialPortStopBits
  - \_quickSpin, [138](#)
- SerialPortStopBits\_Bits1
  - CameraDefsC.h, [291](#)
- SerialPortStopBits\_Bits1AndAHalf
  - CameraDefsC.h, [291](#)
- SerialPortStopBits\_Bits2



- CameraDefsC.h, [291](#)
- SerialReceiveFramingErrorCount
  - \_quickSpin, [138](#)
- SerialReceiveParityErrorCount
  - \_quickSpin, [138](#)
- SerialReceiveQueueClear
  - \_quickSpin, [138](#)
- SerialReceiveQueueCurrentCharacterCount
  - \_quickSpin, [138](#)
- SerialReceiveQueueMaxCharacterCount
  - \_quickSpin, [138](#)
- SerialTransmitQueueCurrentCharacterCount
  - \_quickSpin, [139](#)
- SerialTransmitQueueMaxCharacterCount
  - \_quickSpin, [139](#)
- Sharpening
  - \_quickSpin, [139](#)
- SharpeningAuto
  - \_quickSpin, [139](#)
- SharpeningEnable
  - \_quickSpin, [139](#)
- SharpeningThreshold
  - \_quickSpin, [139](#)
- Signed
  - SpinnakerGenApiDefsC.h, [462](#)
- SoftwareSignalPulse
  - \_quickSpin, [139](#)
- SoftwareSignalSelector
  - \_quickSpin, [139](#)
- SoftwareSignalSelector\_SoftwareSignal0
  - CameraDefsC.h, [291](#)
- SoftwareSignalSelector\_SoftwareSignal1
  - CameraDefsC.h, [291](#)
- SoftwareSignalSelector\_SoftwareSignal2
  - CameraDefsC.h, [291](#)
- SourceCount
  - \_quickSpin, [140](#)
- SourceSelector
  - \_quickSpin, [140](#)
- SourceSelector\_All
  - CameraDefsC.h, [292](#)
- SourceSelector\_Source0
  - CameraDefsC.h, [292](#)
- SourceSelector\_Source1
  - CameraDefsC.h, [292](#)
- SourceSelector\_Source2
  - CameraDefsC.h, [292](#)
- spinArrivalEventFunction
  - SpinnakerDefsC.h, [404](#)
- spinBooleanGetValue
  - SpinnakerGenApiC.h, [418](#)
- spinBooleanSetValue
  - SpinnakerGenApiC.h, [419](#)
- spinCamera
  - SpinnakerDefsC.h, [405](#)
- spinCameraBeginAcquisition
  - SpinnakerC.h, [316](#)
- spinCameraDelInit
  - SpinnakerC.h, [317](#)
- spinCameraDiscoverMaxPacketSize
  - SpinnakerC.h, [317](#)
- spinCameraEndAcquisition
  - SpinnakerC.h, [318](#)
- spinCameraForcelP
  - SpinnakerC.h, [318](#)
- spinCameraGetAccessMode
  - SpinnakerC.h, [318](#)
- spinCameraGetGuiXml
  - SpinnakerC.h, [319](#)
- spinCameraGetNextImage
  - SpinnakerC.h, [319](#)
- spinCameraGetNextImageEx
  - SpinnakerC.h, [320](#)
- spinCameraGetNodeMap
  - SpinnakerC.h, [320](#)
- spinCameraGetTLDeviceNodeMap
  - SpinnakerC.h, [321](#)
- spinCameraGetTLStreamNodeMap
  - SpinnakerC.h, [321](#)
- spinCameraGetUniqueID
  - SpinnakerC.h, [322](#)
- spinCameraInit
  - SpinnakerC.h, [322](#)
- spinCamerasInitialized
  - SpinnakerC.h, [323](#)
- spinCamerasStreaming
  - SpinnakerC.h, [323](#)
- spinCamerasValid
  - SpinnakerC.h, [324](#)
- spinCameraList
  - SpinnakerDefsC.h, [405](#)
- spinCameraListAppend
  - SpinnakerC.h, [324](#)
- spinCameraListClear
  - SpinnakerC.h, [325](#)
- spinCameraListCreateEmpty
  - SpinnakerC.h, [325](#)
- spinCameraListDestroy
  - SpinnakerC.h, [325](#)
- spinCameraListGet
  - SpinnakerC.h, [326](#)
- spinCameraListGetBySerial
  - SpinnakerC.h, [326](#)
- spinCameraListGetSize
  - SpinnakerC.h, [327](#)
- spinCameraListRemove
  - SpinnakerC.h, [327](#)
- spinCameraListRemoveBySerial
  - SpinnakerC.h, [328](#)
- spinCameraReadPort
  - SpinnakerC.h, [328](#)
- spinCameraRegisterDeviceEventHandler
  - SpinnakerC.h, [329](#)
- spinCameraRegisterDeviceEventHandlerEx
  - SpinnakerC.h, [329](#)
- spinCameraRegisterImageEventHandler

- SpinnakerC.h, [330](#)
- spinCameraRelease
  - SpinnakerC.h, [330](#)
- spinCameraUnregisterDeviceEventHandler
  - SpinnakerC.h, [330](#)
- spinCameraUnregisterImageEventHandler
  - SpinnakerC.h, [331](#)
- spinCameraWritePort
  - SpinnakerC.h, [331](#)
- spinCategoryGetFeatureByIndex
  - SpinnakerGenApiC.h, [419](#)
- spinCategoryGetNumFeatures
  - SpinnakerGenApiC.h, [420](#)
- spinCommandExecute
  - SpinnakerGenApiC.h, [420](#)
- spinCommandIsDone
  - SpinnakerGenApiC.h, [420](#)
- spinDeviceArrivalEventHandler
  - SpinnakerDefsC.h, [405](#)
- spinDeviceArrivalEventHandlerCreate
  - SpinnakerC.h, [332](#)
- spinDeviceArrivalEventHandlerDestroy
  - SpinnakerC.h, [332](#)
- spinDeviceEventData
  - SpinnakerDefsC.h, [405](#)
- spinDeviceEventFunction
  - SpinnakerDefsC.h, [405](#)
- spinDeviceEventGetId
  - SpinnakerC.h, [333](#)
- spinDeviceEventGetName
  - SpinnakerC.h, [333](#)
- spinDeviceEventGetPayloadData
  - SpinnakerC.h, [334](#)
- spinDeviceEventGetPayloadDataSize
  - SpinnakerC.h, [334](#)
- spinDeviceEventHandler
  - SpinnakerDefsC.h, [405](#)
- spinDeviceEventHandlerCreate
  - SpinnakerC.h, [335](#)
- spinDeviceEventHandlerDestroy
  - SpinnakerC.h, [335](#)
- spinDeviceRemovalEventHandler
  - SpinnakerDefsC.h, [406](#)
- spinDeviceRemovalEventHandlerCreate
  - SpinnakerC.h, [336](#)
- spinDeviceRemovalEventHandlerDestroy
  - SpinnakerC.h, [336](#)
- spinEnumerationEntryGetEnumValue
  - SpinnakerGenApiC.h, [421](#)
- spinEnumerationEntryGetIntValue
  - SpinnakerGenApiC.h, [421](#)
- spinEnumerationEntryGetSymbolic
  - SpinnakerGenApiC.h, [422](#)
- spinEnumerationGetCurrentEntry
  - SpinnakerGenApiC.h, [422](#)
- spinEnumerationGetEntryByIndex
  - SpinnakerGenApiC.h, [423](#)
- spinEnumerationGetEntryByName
  - SpinnakerGenApiC.h, [423](#)
- spinEnumerationGetNumEntries
  - SpinnakerGenApiC.h, [424](#)
- spinEnumerationSetEnumValue
  - SpinnakerGenApiC.h, [424](#)
- spinEnumerationSetIntValue
  - SpinnakerGenApiC.h, [425](#)
- spinErrorGetLast
  - SpinnakerC.h, [337](#)
- spinErrorGetLastBuildDate
  - SpinnakerC.h, [337](#)
- spinErrorGetLastBuildTime
  - SpinnakerC.h, [337](#)
- spinErrorGetLastFileName
  - SpinnakerC.h, [338](#)
- spinErrorGetLastFullMessage
  - SpinnakerC.h, [338](#)
- spinErrorGetLastFunctionName
  - SpinnakerC.h, [339](#)
- spinErrorGetLastLineNumber
  - SpinnakerC.h, [339](#)
- spinErrorGetLastMessage
  - SpinnakerC.h, [340](#)
- spinFloatGetMax
  - SpinnakerGenApiC.h, [425](#)
- spinFloatGetMin
  - SpinnakerGenApiC.h, [426](#)
- spinFloatGetRepresentation
  - SpinnakerGenApiC.h, [426](#)
- spinFloatGetUnit
  - SpinnakerGenApiC.h, [427](#)
- spinFloatGetValue
  - SpinnakerGenApiC.h, [427](#)
- spinFloatGetValueEx
  - SpinnakerGenApiC.h, [428](#)
- spinFloatSetValue
  - SpinnakerGenApiC.h, [428](#)
- spinFloatSetValueEx
  - SpinnakerGenApiC.h, [429](#)
- spinImage
  - SpinnakerDefsC.h, [406](#)
- spinImageCalculateStatistics
  - SpinnakerC.h, [340](#)
- spinImageCheckCRC
  - SpinnakerC.h, [341](#)
- spinImageChunkDataGetFloatValue
  - SpinnakerC.h, [341](#)
- spinImageChunkDataGetIntValue
  - SpinnakerC.h, [342](#)
- spinImageConvert
  - SpinnakerC.h, [342](#)
- spinImageConvertEx
  - SpinnakerC.h, [342](#)
- spinImageCreate
  - SpinnakerC.h, [343](#)
- spinImageCreateEmpty
  - SpinnakerC.h, [343](#)
- spinImageCreateEx



- SpinnakerC.h, [344](#)
- spinImageDeepCopy
  - SpinnakerC.h, [344](#)
- spinImageDestroy
  - SpinnakerC.h, [345](#)
- spinImageEventFunction
  - SpinnakerDefsC.h, [406](#)
- spinImageEventHandler
  - SpinnakerDefsC.h, [406](#)
- spinImageEventHandlerCreate
  - SpinnakerC.h, [345](#)
- spinImageEventHandlerDestroy
  - SpinnakerC.h, [346](#)
- spinImageGetBitsPerPixel
  - SpinnakerC.h, [346](#)
- spinImageGetBufferSize
  - SpinnakerC.h, [347](#)
- spinImageGetChunkLayoutID
  - SpinnakerC.h, [347](#)
- spinImageGetColorProcessing
  - SpinnakerC.h, [348](#)
- spinImageGetData
  - SpinnakerC.h, [348](#)
- spinImageGetDefaultColorProcessing
  - SpinnakerC.h, [349](#)
- spinImageGetFrameID
  - SpinnakerC.h, [349](#)
- spinImageGetHeight
  - SpinnakerC.h, [350](#)
- spinImageGetID
  - SpinnakerC.h, [350](#)
- spinImageGetOffsetX
  - SpinnakerC.h, [351](#)
- spinImageGetOffsetY
  - SpinnakerC.h, [351](#)
- spinImageGetPaddingX
  - SpinnakerC.h, [352](#)
- spinImageGetPaddingY
  - SpinnakerC.h, [352](#)
- spinImageGetPayloadType
  - SpinnakerC.h, [353](#)
- spinImageGetPixelFormat
  - SpinnakerC.h, [353](#)
- spinImageGetPixelFormatName
  - SpinnakerC.h, [354](#)
- spinImageGetPrivateData
  - SpinnakerC.h, [354](#)
- spinImageGetSize
  - SpinnakerC.h, [355](#)
- spinImageGetStatus
  - SpinnakerC.h, [355](#)
- spinImageGetStatusDescription
  - SpinnakerC.h, [356](#)
- spinImageGetStride
  - SpinnakerC.h, [356](#)
- spinImageGetTimeStamp
  - SpinnakerC.h, [357](#)
- spinImageGetTLPayloadType
  - SpinnakerC.h, [357](#)
- spinImageGetTLPixelFormat
  - SpinnakerC.h, [358](#)
- spinImageGetTLPixelFormatNamespace
  - SpinnakerC.h, [358](#)
- spinImageGetValidPayloadSize
  - SpinnakerC.h, [359](#)
- spinImageGetWidth
  - SpinnakerC.h, [359](#)
- spinImageHasCRC
  - SpinnakerC.h, [360](#)
- spinImageIsIncomplete
  - SpinnakerC.h, [360](#)
- spinImageRelease
  - SpinnakerC.h, [361](#)
- spinImageReset
  - SpinnakerC.h, [361](#)
- spinImageResetEx
  - SpinnakerC.h, [362](#)
- spinImageSave
  - SpinnakerC.h, [362](#)
- spinImageSaveBmp
  - SpinnakerC.h, [363](#)
- spinImageSaveFromExt
  - SpinnakerC.h, [363](#)
- spinImageSaveJpeg
  - SpinnakerC.h, [364](#)
- spinImageSaveJpg2
  - SpinnakerC.h, [364](#)
- spinImageSavePgm
  - SpinnakerC.h, [365](#)
- spinImageSavePng
  - SpinnakerC.h, [365](#)
- spinImageSavePpm
  - SpinnakerC.h, [366](#)
- spinImageSaveTiff
  - SpinnakerC.h, [366](#)
- spinImageSetDefaultColorProcessing
  - SpinnakerC.h, [367](#)
- spinImageStatistics
  - SpinnakerDefsC.h, [406](#)
- spinImageStatisticsCreate
  - SpinnakerC.h, [367](#)
- spinImageStatisticsDestroy
  - SpinnakerC.h, [368](#)
- spinImageStatisticsDisableAll
  - SpinnakerC.h, [368](#)
- spinImageStatisticsEnableAll
  - SpinnakerC.h, [368](#)
- spinImageStatisticsEnableGreyOnly
  - SpinnakerC.h, [369](#)
- spinImageStatisticsEnableHslOnly
  - SpinnakerC.h, [369](#)
- spinImageStatisticsEnableRgbOnly
  - SpinnakerC.h, [370](#)
- spinImageStatisticsGetAll
  - SpinnakerC.h, [370](#)
- spinImageStatisticsGetChannelStatus

- SpinnakerC.h, [371](#)
- spinImageStatisticsGetHistogram
  - SpinnakerC.h, [371](#)
- spinImageStatisticsGetMean
  - SpinnakerC.h, [372](#)
- spinImageStatisticsGetNumPixelValues
  - SpinnakerC.h, [372](#)
- spinImageStatisticsGetPixelValueRange
  - SpinnakerC.h, [373](#)
- spinImageStatisticsGetRange
  - SpinnakerC.h, [373](#)
- spinImageStatisticsSetChannelStatus
  - SpinnakerC.h, [374](#)
- spinIntegerGetInc
  - SpinnakerGenApiC.h, [429](#)
- spinIntegerGetMax
  - SpinnakerGenApiC.h, [430](#)
- spinIntegerGetMin
  - SpinnakerGenApiC.h, [430](#)
- spinIntegerGetRepresentation
  - SpinnakerGenApiC.h, [431](#)
- spinIntegerGetValue
  - SpinnakerGenApiC.h, [431](#)
- spinIntegerGetValueEx
  - SpinnakerGenApiC.h, [432](#)
- spinIntegerSetValue
  - SpinnakerGenApiC.h, [432](#)
- spinIntegerSetValueEx
  - SpinnakerGenApiC.h, [433](#)
- spinInterface
  - SpinnakerDefsC.h, [406](#)
- spinInterfaceEventHandler
  - SpinnakerDefsC.h, [407](#)
- spinInterfaceEventHandlerCreate
  - SpinnakerC.h, [374](#)
- spinInterfaceEventHandlerDestroy
  - SpinnakerC.h, [375](#)
- spinInterfaceGetCameras
  - SpinnakerC.h, [375](#)
- spinInterfaceGetCamerasEx
  - SpinnakerC.h, [376](#)
- spinInterfaceGetTLNodeMap
  - SpinnakerC.h, [376](#)
- spinInterfaceIsInUse
  - SpinnakerC.h, [377](#)
- spinInterfaceList
  - SpinnakerDefsC.h, [407](#)
- spinInterfaceListClear
  - SpinnakerC.h, [377](#)
- spinInterfaceListCreateEmpty
  - SpinnakerC.h, [378](#)
- spinInterfaceListDestroy
  - SpinnakerC.h, [378](#)
- spinInterfaceListGet
  - SpinnakerC.h, [379](#)
- spinInterfaceListGetSize
  - SpinnakerC.h, [379](#)
- spinInterfaceRegisterDeviceArrivalEventHandler
  - SpinnakerC.h, [380](#)
- spinInterfaceRegisterDeviceRemovalEventHandler
  - SpinnakerC.h, [380](#)
- spinInterfaceRegisterInterfaceEventHandler
  - SpinnakerC.h, [381](#)
- spinInterfaceRelease
  - SpinnakerC.h, [381](#)
- spinInterfaceSendActionCommand
  - SpinnakerC.h, [381](#)
- spinInterfaceUnregisterDeviceArrivalEventHandler
  - SpinnakerC.h, [382](#)
- spinInterfaceUnregisterDeviceRemovalEventHandler
  - SpinnakerC.h, [383](#)
- spinInterfaceUnregisterInterfaceEventHandler
  - SpinnakerC.h, [383](#)
- spinInterfaceUpdateCameras
  - SpinnakerC.h, [384](#)
- spinLogDataGetCategoryName
  - SpinnakerC.h, [384](#)
- spinLogDataGetLogMessage
  - SpinnakerC.h, [385](#)
- spinLogDataGetNDC
  - SpinnakerC.h, [385](#)
- spinLogDataGetPriority
  - SpinnakerC.h, [386](#)
- spinLogDataGetPriorityName
  - SpinnakerC.h, [386](#)
- spinLogDataGetThreadName
  - SpinnakerC.h, [387](#)
- spinLogDataGetTimestamp
  - SpinnakerC.h, [387](#)
- spinLogEventData
  - SpinnakerDefsC.h, [407](#)
- spinLogEventFunction
  - SpinnakerDefsC.h, [407](#)
- spinLogEventHandler
  - SpinnakerDefsC.h, [407](#)
- spinLogEventHandlerCreate
  - SpinnakerC.h, [388](#)
- spinLogEventHandlerDestroy
  - SpinnakerC.h, [388](#)
- Spinnaker C API, [12](#)
- Spinnaker C Definitions, [7](#)
- Spinnaker C Enumerations, [27](#)
- Spinnaker C Function Signatures, [26](#)
- Spinnaker C GenICam API, [29](#)
- Spinnaker C GenICam Enumerations, [43](#)
- Spinnaker C GenICam Handles, [42](#)
- Spinnaker C Handles, [25](#)
- Spinnaker C QuickSpin API, [10](#)
- Spinnaker C Structures, [28](#)
- SPINNAKER\_ERR\_ABORT
  - SpinnakerDefsC.h, [409](#)
- SPINNAKER\_ERR\_ACCESS\_DENIED
  - SpinnakerDefsC.h, [409](#)
- SPINNAKER\_ERR\_BUFFER\_TOO\_SMALL
  - SpinnakerDefsC.h, [410](#)
- SPINNAKER\_ERR\_BUSY

- SpinnakerDefsC.h, [410](#)
- SPINNAKER\_ERR\_CUSTOM\_ID
  - SpinnakerDefsC.h, [410](#)
- SPINNAKER\_ERR\_ERROR
  - SpinnakerDefsC.h, [409](#)
- SPINNAKER\_ERR\_IM\_COLOR\_CONVERSION
  - SpinnakerDefsC.h, [410](#)
- SPINNAKER\_ERR\_IM\_CONVERT
  - SpinnakerDefsC.h, [410](#)
- SPINNAKER\_ERR\_IM\_COPY
  - SpinnakerDefsC.h, [410](#)
- SPINNAKER\_ERR\_IM\_HISTOGRAM\_MEAN
  - SpinnakerDefsC.h, [410](#)
- SPINNAKER\_ERR\_IM\_HISTOGRAM\_RANGE
  - SpinnakerDefsC.h, [410](#)
- SPINNAKER\_ERR\_IM\_MALLOC
  - SpinnakerDefsC.h, [410](#)
- SPINNAKER\_ERR\_IM\_MIN\_MAX
  - SpinnakerDefsC.h, [410](#)
- SPINNAKER\_ERR\_IM\_NOT\_SUPPORTED
  - SpinnakerDefsC.h, [410](#)
- SPINNAKER\_ERR\_INVALID\_ADDRESS
  - SpinnakerDefsC.h, [410](#)
- SPINNAKER\_ERR\_INVALID\_BUFFER
  - SpinnakerDefsC.h, [409](#)
- SPINNAKER\_ERR\_INVALID\_HANDLE
  - SpinnakerDefsC.h, [409](#)
- SPINNAKER\_ERR\_INVALID\_ID
  - SpinnakerDefsC.h, [409](#)
- SPINNAKER\_ERR\_INVALID\_INDEX
  - SpinnakerDefsC.h, [410](#)
- SPINNAKER\_ERR\_INVALID\_PARAMETER
  - SpinnakerDefsC.h, [409](#)
- SPINNAKER\_ERR\_INVALID\_VALUE
  - SpinnakerDefsC.h, [410](#)
- SPINNAKER\_ERR\_IO
  - SpinnakerDefsC.h, [409](#)
- SPINNAKER\_ERR\_NO\_DATA
  - SpinnakerDefsC.h, [409](#)
- SPINNAKER\_ERR\_NOT\_AVAILABLE
  - SpinnakerDefsC.h, [410](#)
- SPINNAKER\_ERR\_NOT\_IMPLEMENTED
  - SpinnakerDefsC.h, [409](#)
- SPINNAKER\_ERR\_NOT\_INITIALIZED
  - SpinnakerDefsC.h, [409](#)
- SPINNAKER\_ERR\_OUT\_OF\_MEMORY
  - SpinnakerDefsC.h, [410](#)
- SPINNAKER\_ERR\_PARSING\_CHUNK\_DATA
  - SpinnakerDefsC.h, [410](#)
- SPINNAKER\_ERR\_RESOURCE\_EXHAUSTED
  - SpinnakerDefsC.h, [410](#)
- SPINNAKER\_ERR\_RESOURCE\_IN\_USE
  - SpinnakerDefsC.h, [409](#)
- SPINNAKER\_ERR\_SUCCESS
  - SpinnakerDefsC.h, [409](#)
- SPINNAKER\_ERR\_TIMEOUT
  - SpinnakerDefsC.h, [409](#)
- SPINNAKER\_PIXELFORMAT\_NAMESPACE\_CUSTOM\_ID
  - SpinnakerDefsC.h, [413](#)
- SPINNAKER\_PIXELFORMAT\_NAMESPACE\_GEV
  - SpinnakerDefsC.h, [413](#)
- SPINNAKER\_PIXELFORMAT\_NAMESPACE\_IIDC
  - SpinnakerDefsC.h, [413](#)
- SPINNAKER\_PIXELFORMAT\_NAMESPACE\_PFNC\_16BIT
  - SpinnakerDefsC.h, [413](#)
- SPINNAKER\_PIXELFORMAT\_NAMESPACE\_PFNC\_32BIT
  - SpinnakerDefsC.h, [413](#)
- SPINNAKER\_PIXELFORMAT\_NAMESPACE\_UNKNOWN
  - SpinnakerDefsC.h, [413](#)
- SpinnakerC.h
  - spinCameraBeginAcquisition, [316](#)
  - spinCameraDelInit, [317](#)
  - spinCameraDiscoverMaxPacketSize, [317](#)
  - spinCameraEndAcquisition, [318](#)
  - spinCameraForceIP, [318](#)
  - spinCameraGetAccessMode, [318](#)
  - spinCameraGetGuiXml, [319](#)
  - spinCameraGetNextImage, [319](#)
  - spinCameraGetNextImageEx, [320](#)
  - spinCameraGetNodeMap, [320](#)
  - spinCameraGetTLDeviceNodeMap, [321](#)
  - spinCameraGetTLStreamNodeMap, [321](#)
  - spinCameraGetUniqueID, [322](#)
  - spinCameraInit, [322](#)
  - spinCameraIsInitialized, [323](#)
  - spinCameraIsStreaming, [323](#)
  - spinCameraIsValid, [324](#)
  - spinCameraListAppend, [324](#)
  - spinCameraListClear, [325](#)
  - spinCameraListCreateEmpty, [325](#)
  - spinCameraListDestroy, [325](#)
  - spinCameraListGet, [326](#)
  - spinCameraListGetBySerial, [326](#)
  - spinCameraListGetSize, [327](#)
  - spinCameraListRemove, [327](#)
  - spinCameraListRemoveBySerial, [328](#)
  - spinCameraReadPort, [328](#)
  - spinCameraRegisterDeviceEventHandler, [329](#)
  - spinCameraRegisterDeviceEventHandlerEx, [329](#)
  - spinCameraRegisterImageEventHandler, [330](#)
  - spinCameraRelease, [330](#)
  - spinCameraUnregisterDeviceEventHandler, [330](#)
  - spinCameraUnregisterImageEventHandler, [331](#)
  - spinCameraWritePort, [331](#)
  - spinDeviceArrivalEventHandlerCreate, [332](#)
  - spinDeviceArrivalEventHandlerDestroy, [332](#)
  - spinDeviceEventGetId, [333](#)
  - spinDeviceEventGetName, [333](#)
  - spinDeviceEventGetPayloadData, [334](#)
  - spinDeviceEventGetPayloadDataSize, [334](#)
  - spinDeviceEventHandlerCreate, [335](#)
  - spinDeviceEventHandlerDestroy, [335](#)
  - spinDeviceRemovalEventHandlerCreate, [336](#)
  - spinDeviceRemovalEventHandlerDestroy, [336](#)
  - spinErrorGetLast, [337](#)
  - spinErrorGetLastBuildDate, [337](#)

- spinErrorGetLastBuildTime, 337
- spinErrorGetLastFileName, 338
- spinErrorGetLastFullMessage, 338
- spinErrorGetLastFunctionName, 339
- spinErrorGetLastLineNumber, 339
- spinErrorGetLastMessage, 340
- spinImageCalculateStatistics, 340
- spinImageCheckCRC, 341
- spinImageChunkDataGetFloatValue, 341
- spinImageChunkDataGetIntValue, 342
- spinImageConvert, 342
- spinImageConvertEx, 342
- spinImageCreate, 343
- spinImageCreateEmpty, 343
- spinImageCreateEx, 344
- spinImageDeepCopy, 344
- spinImageDestroy, 345
- spinImageEventHandlerCreate, 345
- spinImageEventHandlerDestroy, 346
- spinImageGetBitsPerPixel, 346
- spinImageGetBufferSize, 347
- spinImageGetChunkLayoutID, 347
- spinImageGetColorProcessing, 348
- spinImageGetData, 348
- spinImageGetDefaultColorProcessing, 349
- spinImageGetFrameID, 349
- spinImageGetHeight, 350
- spinImageGetID, 350
- spinImageGetOffsetX, 351
- spinImageGetOffsetY, 351
- spinImageGetPaddingX, 352
- spinImageGetPaddingY, 352
- spinImageGetPayloadType, 353
- spinImageGetPixelFormat, 353
- spinImageGetPixelFormatName, 354
- spinImageGetPrivateData, 354
- spinImageGetSize, 355
- spinImageGetStatus, 355
- spinImageGetStatusDescription, 356
- spinImageGetStride, 356
- spinImageGetTimeStamp, 357
- spinImageGetTLPayloadType, 357
- spinImageGetTLPixelFormat, 358
- spinImageGetTLPixelFormatNamespace, 358
- spinImageGetValidPayloadSize, 359
- spinImageGetWidth, 359
- spinImageHasCRC, 360
- spinImageIsIncomplete, 360
- spinImageRelease, 361
- spinImageReset, 361
- spinImageResetEx, 362
- spinImageSave, 362
- spinImageSaveBmp, 363
- spinImageSaveFromExt, 363
- spinImageSaveJpeg, 364
- spinImageSaveJpg2, 364
- spinImageSavePgm, 365
- spinImageSavePng, 365
- spinImageSavePpm, 366
- spinImageSaveTiff, 366
- spinImageSetDefaultColorProcessing, 367
- spinImageStatisticsCreate, 367
- spinImageStatisticsDestroy, 368
- spinImageStatisticsDisableAll, 368
- spinImageStatisticsEnableAll, 368
- spinImageStatisticsEnableGreyOnly, 369
- spinImageStatisticsEnableHslOnly, 369
- spinImageStatisticsEnableRgbOnly, 370
- spinImageStatisticsGetAll, 370
- spinImageStatisticsGetChannelStatus, 371
- spinImageStatisticsGetHistogram, 371
- spinImageStatisticsGetMean, 372
- spinImageStatisticsGetNumPixelValues, 372
- spinImageStatisticsGetPixelValueRange, 373
- spinImageStatisticsGetRange, 373
- spinImageStatisticsSetChannelStatus, 374
- spinInterfaceEventHandlerCreate, 374
- spinInterfaceEventHandlerDestroy, 375
- spinInterfaceGetCameras, 375
- spinInterfaceGetCamerasEx, 376
- spinInterfaceGetTLNodeMap, 376
- spinInterfaceIsInUse, 377
- spinInterfaceListClear, 377
- spinInterfaceListCreateEmpty, 378
- spinInterfaceListDestroy, 378
- spinInterfaceListGet, 379
- spinInterfaceListGetSize, 379
- spinInterfaceRegisterDeviceArrivalEventHandler, 380
- spinInterfaceRegisterDeviceRemovalEventHandler, 380
- spinInterfaceRegisterInterfaceEventHandler, 381
- spinInterfaceRelease, 381
- spinInterfaceSendActionCommand, 381
- spinInterfaceUnregisterDeviceArrivalEventHandler, 382
- spinInterfaceUnregisterDeviceRemovalEventHandler, 383
- spinInterfaceUnregisterInterfaceEventHandler, 383
- spinInterfaceUpdateCameras, 384
- spinLogDataGetCategoryName, 384
- spinLogDataGetLogMessage, 385
- spinLogDataGetNDC, 385
- spinLogDataGetPriority, 386
- spinLogDataGetPriorityName, 386
- spinLogDataGetThreadName, 387
- spinLogDataGetTimestamp, 387
- spinLogEventHandlerCreate, 388
- spinLogEventHandlerDestroy, 388
- spinSystemGetCameras, 389
- spinSystemGetCamerasEx, 389
- spinSystemGetInstance, 390
- spinSystemGetInterfaces, 390
- spinSystemGetLibraryVersion, 391
- spinSystemGetLoggingLevel, 391
- spinSystemGetTLNodeMap, 391

- spinSystemIsInUse, 392
- spinSystemRegisterDeviceArrivalEventHandler, 392
- spinSystemRegisterDeviceRemovalEventHandler, 393
- spinSystemRegisterInterfaceEventHandler, 393
- spinSystemRegisterLogEventHandler, 394
- spinSystemReleaseInstance, 394
- spinSystemSendActionCommand, 395
- spinSystemSetLoggingLevel, 396
- spinSystemUnregisterAllLogEventHandlers, 396
- spinSystemUnregisterDeviceArrivalEventHandler, 396
- spinSystemUnregisterDeviceRemovalEventHandler, 397
- spinSystemUnregisterInterfaceEventHandler, 397
- spinSystemUnregisterLogEventHandler, 398
- spinSystemUpdateCameras, 398
- spinSystemUpdateCamerasEx, 399
- SPINNAKERC\_API
  - SpinnakerPlatformC.h, 464
- SpinnakerDefsC.h
  - \_actionCommandStatus, 408
  - \_spinColorProcessingAlgorithm, 408
  - \_spinError, 409
  - \_spinImageFileFormat, 410
  - \_spinImageStatus, 411
  - \_spinLogLevel, 411
  - \_spinPayloadTypeInfoIDs, 412
  - \_spinPixelFormatNamespaceID, 412
  - \_spinStatisticsChannel, 413
- ACTION\_COMMAND\_STATUS\_ACTION\_LATE, 408
- ACTION\_COMMAND\_STATUS\_ERROR, 408
- ACTION\_COMMAND\_STATUS\_NO\_REF\_TIME, 408
- ACTION\_COMMAND\_STATUS\_OK, 408
- ACTION\_COMMAND\_STATUS\_OVERFLOW, 408
- ADOBE\_DEFLATE, 413
- BILINEAR, 409
- BLUE, 413
- BMP, 410
- bool8\_t, 404
- CCITTFAX3, 413
- CCITTFAX4, 413
- CompressionMethod, 413
- DEFAULT, 409
- DEFLATE, 413
- DIRECTIONAL\_FILTER, 409
- EDGE\_SENSING, 409
- False, 414
- FROM\_FILE\_EXT, 410
- GENICAM\_ERR\_ACCESS, 410
- GENICAM\_ERR\_BAD\_ALLOCATION, 410
- GENICAM\_ERR\_DYNAMIC\_CAST, 410
- GENICAM\_ERR\_GENERIC, 410
- GENICAM\_ERR\_INVALID\_ARGUMENT, 410
- GENICAM\_ERR\_LOGICAL, 410
- GENICAM\_ERR\_OUT\_OF\_RANGE, 410
- GENICAM\_ERR\_PROPERTY, 410
- GENICAM\_ERR\_RUN\_TIME, 410
- GENICAM\_ERR\_TIMEOUT, 410
- GREEN, 413
- GREY, 413
- HQ\_LINEAR, 409
- HUE, 413
- IMAGE\_CHUNK\_DATA\_INVALID, 411
- IMAGE\_CRC\_CHECK\_FAILED, 411
- IMAGE\_DATA\_INCOMPLETE, 411
- IMAGE\_DATA\_OVERFLOW, 411
- IMAGE\_FILE\_FORMAT\_FORCE\_32BITS, 411
- IMAGE\_INFO\_INCONSISTENT, 411
- IMAGE\_LEADER\_BUFFER\_SIZE\_INCONSISTENT, 411
- IMAGE\_MISSING\_LEADER, 411
- IMAGE\_MISSING\_PACKETS, 411
- IMAGE\_MISSING\_TRAILER, 411
- IMAGE\_NO\_ERROR, 411
- IMAGE\_NO\_SYSTEM\_RESOURCES, 411
- IMAGE\_PACKETID\_INCONSISTENT, 411
- IMAGE\_TRAILER\_BUFFER\_SIZE\_INCONSISTENT, 411
- IMAGE\_UNKNOWN\_ERROR, 411
- IPP, 409
- JPEG, 410
- JPEG2000, 410
- JPG, 413
- LIGHTNESS, 413
- LOG\_LEVEL\_ALERT, 412
- LOG\_LEVEL\_CRIT, 412
- LOG\_LEVEL\_DEBUG, 412
- LOG\_LEVEL\_ERROR, 412
- LOG\_LEVEL\_FATAL, 412
- LOG\_LEVEL\_INFO, 412
- LOG\_LEVEL\_NOTICE, 412
- LOG\_LEVEL\_NOTSET, 412
- LOG\_LEVEL\_OFF, 412
- LOG\_LEVEL\_WARN, 412
- LZW, 413
- NEAREST\_NEIGHBOR, 409
- NEAREST\_NEIGHBOR\_AVG, 409
- NO\_COLOR\_PROCESSING, 409
- NONE, 413
- NUM\_STATISTICS\_CHANNELS, 413
- PACKBITS, 413
- PAYLOAD\_TYPE\_CHUNK\_DATA, 412
- PAYLOAD\_TYPE\_CHUNK\_ONLY, 412
- PAYLOAD\_TYPE\_CUSTOM\_ID, 412
- PAYLOAD\_TYPE\_DEVICE\_SPECIFIC, 412
- PAYLOAD\_TYPE\_EXTENDED\_CHUNK, 412
- PAYLOAD\_TYPE\_FILE, 412
- PAYLOAD\_TYPE\_H264, 412
- PAYLOAD\_TYPE\_IMAGE, 412
- PAYLOAD\_TYPE\_JPEG, 412
- PAYLOAD\_TYPE\_JPEG2000, 412
- PAYLOAD\_TYPE\_MULTI\_PART, 412

- PAYLOAD\_TYPE\_RAW\_DATA, [412](#)
- PAYLOAD\_TYPE\_UNKNOWN, [412](#)
- PGM, [410](#)
- PNG, [411](#)
- PPM, [410](#)
- RAW, [411](#)
- RED, [413](#)
- RIGOROUS, [409](#)
- SATURATION, [413](#)
- spinArrivalEventFunction, [404](#)
- spinCamera, [405](#)
- spinCameraList, [405](#)
- spinDeviceArrivalEventHandler, [405](#)
- spinDeviceEventData, [405](#)
- spinDeviceEventFunction, [405](#)
- spinDeviceEventHandler, [405](#)
- spinDeviceRemovalEventHandler, [406](#)
- spinImage, [406](#)
- spinImageEventFunction, [406](#)
- spinImageEventHandler, [406](#)
- spinImageStatistics, [406](#)
- spinInterface, [406](#)
- spinInterfaceEventHandler, [407](#)
- spinInterfaceList, [407](#)
- spinLogEventData, [407](#)
- spinLogEventFunction, [407](#)
- spinLogEventHandler, [407](#)
- SPINNAKER\_ERR\_ABORT, [409](#)
- SPINNAKER\_ERR\_ACCESS\_DENIED, [409](#)
- SPINNAKER\_ERR\_BUFFER\_TOO\_SMALL, [410](#)
- SPINNAKER\_ERR\_BUSY, [410](#)
- SPINNAKER\_ERR\_CUSTOM\_ID, [410](#)
- SPINNAKER\_ERR\_ERROR, [409](#)
- SPINNAKER\_ERR\_IM\_COLOR\_CONVERSION, [410](#)
- SPINNAKER\_ERR\_IM\_CONVERT, [410](#)
- SPINNAKER\_ERR\_IM\_COPY, [410](#)
- SPINNAKER\_ERR\_IM\_HISTOGRAM\_MEAN, [410](#)
- SPINNAKER\_ERR\_IM\_HISTOGRAM\_RANGE, [410](#)
- SPINNAKER\_ERR\_IM\_MALLOC, [410](#)
- SPINNAKER\_ERR\_IM\_MIN\_MAX, [410](#)
- SPINNAKER\_ERR\_IM\_NOT\_SUPPORTED, [410](#)
- SPINNAKER\_ERR\_INVALID\_ADDRESS, [410](#)
- SPINNAKER\_ERR\_INVALID\_BUFFER, [409](#)
- SPINNAKER\_ERR\_INVALID\_HANDLE, [409](#)
- SPINNAKER\_ERR\_INVALID\_ID, [409](#)
- SPINNAKER\_ERR\_INVALID\_INDEX, [410](#)
- SPINNAKER\_ERR\_INVALID\_PARAMETER, [409](#)
- SPINNAKER\_ERR\_INVALID\_VALUE, [410](#)
- SPINNAKER\_ERR\_IO, [409](#)
- SPINNAKER\_ERR\_NO\_DATA, [409](#)
- SPINNAKER\_ERR\_NOT\_AVAILABLE, [410](#)
- SPINNAKER\_ERR\_NOT\_IMPLEMENTED, [409](#)
- SPINNAKER\_ERR\_NOT\_INITIALIZED, [409](#)
- SPINNAKER\_ERR\_OUT\_OF\_MEMORY, [410](#)
- SPINNAKER\_ERR\_PARSING\_CHUNK\_DATA, [410](#)
- SPINNAKER\_ERR\_RESOURCE\_EXHAUSTED, [410](#)
- SPINNAKER\_ERR\_RESOURCE\_IN\_USE, [409](#)
- SPINNAKER\_ERR\_SUCCESS, [409](#)
- SPINNAKER\_ERR\_TIMEOUT, [409](#)
- SPINNAKER\_PIXELFORMAT\_NAMESPACE\_CUSTOM\_ID, [413](#)
- SPINNAKER\_PIXELFORMAT\_NAMESPACE\_GEV, [413](#)
- SPINNAKER\_PIXELFORMAT\_NAMESPACE\_IIDC, [413](#)
- SPINNAKER\_PIXELFORMAT\_NAMESPACE\_PFNC\_16BIT, [413](#)
- SPINNAKER\_PIXELFORMAT\_NAMESPACE\_PFNC\_32BIT, [413](#)
- SPINNAKER\_PIXELFORMAT\_NAMESPACE\_UNKNOWN, [413](#)
- spinRemovalEventFunction, [407](#)
- spinSystem, [408](#)
- spinVideo, [408](#)
- TIFF, [410](#)
- True, [414](#)
- WEIGHTED\_DIRECTIONAL\_FILTER, [409](#)
- SpinnakerGenApiC.h
  - spinBooleanGetValue, [418](#)
  - spinBooleanSetValue, [419](#)
  - spinCategoryGetFeatureByIndex, [419](#)
  - spinCategoryGetNumFeatures, [420](#)
  - spinCommandExecute, [420](#)
  - spinCommandIsDone, [420](#)
  - spinEnumerationEntryGetEnumValue, [421](#)
  - spinEnumerationEntryGetIntValue, [421](#)
  - spinEnumerationEntryGetSymbolic, [422](#)
  - spinEnumerationGetCurrentEntry, [422](#)
  - spinEnumerationGetEntryByIndex, [423](#)
  - spinEnumerationGetEntryByName, [423](#)
  - spinEnumerationGetNumEntries, [424](#)
  - spinEnumerationSetEnumValue, [424](#)
  - spinEnumerationSetIntValue, [425](#)
  - spinFloatGetMax, [425](#)
  - spinFloatGetMin, [426](#)
  - spinFloatGetRepresentation, [426](#)
  - spinFloatGetUnit, [427](#)
  - spinFloatGetValue, [427](#)
  - spinFloatGetValueEx, [428](#)
  - spinFloatSetValue, [428](#)
  - spinFloatSetValueEx, [429](#)
  - spinIntegerGetInc, [429](#)
  - spinIntegerGetMax, [430](#)
  - spinIntegerGetMin, [430](#)
  - spinIntegerGetRepresentation, [431](#)
  - spinIntegerGetValue, [431](#)
  - spinIntegerGetValueEx, [432](#)
  - spinIntegerSetValue, [432](#)
  - spinIntegerSetValueEx, [433](#)
  - spinNodeDeregisterCallback, [433](#)
  - spinNodeFromString, [434](#)
  - spinNodeFromStringEx, [434](#)



- spinNodeGetAccessMode, 435
- spinNodeGetCachingMode, 435
- spinNodeGetDescription, 436
- spinNodeGetDisplayName, 436
- spinNodeGetImposedAccessMode, 437
- spinNodeGetImposedVisibility, 437
- spinNodeGetName, 438
- spinNodeGetNameSpace, 438
- spinNodeGetPollingTime, 439
- spinNodeGetToolTip, 439
- spinNodeGetType, 440
- spinNodeGetVisibility, 440
- spinNodeInvalidateNode, 441
- spinNodeIsAvailable, 441
- spinNodeIsEqual, 442
- spinNodeIsImplemented, 442
- spinNodeIsReadable, 443
- spinNodeIsWritable, 443
- spinNodeMapGetNode, 444
- spinNodeMapGetNodeByIndex, 444
- spinNodeMapGetNumNodes, 445
- spinNodeMapPoll, 445
- spinNodeRegisterCallback, 446
- spinNodeToString, 446
- spinNodeToStringEx, 447
- spinRegisterGet, 447
- spinRegisterGetAddress, 448
- spinRegisterGetEx, 448
- spinRegisterGetLength, 449
- spinRegisterSet, 449
- spinRegisterSetEx, 450
- spinRegisterSetReference, 450
- spinStringGetMaxLength, 451
- spinStringGetValue, 451
- spinStringGetValueEx, 452
- spinStringSetValue, 452
- spinStringSetValueEx, 453
- SpinnakerGenApiDefsC.h
  - \_CycleDetectAccessMode, 457
  - \_UndefinedAccessMode, 457
  - \_UndefinedCachingMode, 457
  - \_UndefinedEDisplayNotation, 458
  - \_UndefinedESlope, 462
  - \_UndefinedEXMLValidation, 463
  - \_UndefinedEndian, 458
  - \_UndefinedNameSpace, 460
  - \_UndefinedRepresentation, 461
  - \_UndefinedSign, 462
  - \_UndefinedStandardNameSpace, 462
  - \_UndefinedVisibility, 463
  - \_UndefinedYesNo, 464
  - \_spinAccessMode, 457
  - \_spinCachingMode, 457
  - \_spinDisplayNotation, 457
  - \_spinEndianness, 458
  - \_spinIncMode, 458
  - \_spinInputDirection, 458
  - \_spinInterfaceType, 459
  - \_spinLinkType, 460
  - \_spinNameSpace, 460
  - \_spinNodeType, 461
  - \_spinRepresentation, 461
  - \_spinSign, 461
  - \_spinSlope, 462
  - \_spinStandardNameSpace, 462
  - \_spinVisibility, 463
  - \_spinXMLValidation, 463
  - \_spinYesNo, 463
  - Automatic, 462
  - BaseNode, 461
  - Beginner, 463
  - BigEndian, 458
  - Boolean, 461
  - BooleanNode, 461
  - CategoryNode, 461
  - CL, 462
  - CommandNode, 461
  - ctAllDependingNodes, 460
  - ctAllTerminalNodes, 460
  - ctDependingChildren, 460
  - ctInvalidators, 460
  - ctReadingChildren, 460
  - ctWritingChildren, 460
  - Custom, 460
  - Decreasing, 462
  - EnumEntryNode, 461
  - EnumerationNode, 461
  - Expert, 463
  - fixedIncrement, 458
  - FloatNode, 461
  - fnAutomatic, 458
  - fnFixed, 458
  - fnScientific, 458
  - GEV, 462
  - Guru, 463
  - HexNumber, 461
  - idFrom, 459
  - idNone, 459
  - idTo, 459
  - IIDC, 462
  - Increasing, 462
  - IntegerNode, 461
  - intfIBase, 459
  - intfIBoolean, 459
  - intfICategory, 459
  - intfICommand, 459
  - intfIEnumEntry, 459
  - intfIEnumeration, 459
  - intfIFloat, 459
  - intfIInteger, 459
  - intfIPort, 460
  - intfIRegister, 459
  - intfIString, 459
  - intfIValue, 459
  - Invisible, 463
  - IPV4Address, 461

- Linear, [461](#)
- listIncrement, [458](#)
- LittleEndian, [458](#)
- Logarithmic, [461](#)
- MACAddress, [461](#)
- NA, [457](#)
- NI, [457](#)
- No, [464](#)
- NoCache, [457](#)
- noIncrement, [458](#)
- None, [462](#)
- PortNode, [461](#)
- PureNumber, [461](#)
- RegisterNode, [461](#)
- RO, [457](#)
- RW, [457](#)
- Signed, [462](#)
- spinNodeCallbackFunction, [456](#)
- spinNodeCallbackHandle, [456](#)
- spinNodeHandle, [456](#)
- spinNodeMapHandle, [456](#)
- Standard, [460](#)
- StringNode, [461](#)
- UnknownNode, [461](#)
- Unsigned, [462](#)
- USB, [462](#)
- ValueNode, [461](#)
- Varying, [462](#)
- WO, [457](#)
- WriteAround, [457](#)
- WriteThrough, [457](#)
- xvAll, [463](#)
- xvCycles, [463](#)
- xvDefault, [463](#)
- xvLoad, [463](#)
- xvSFNC, [463](#)
- Yes, [464](#)
- SpinnakerPlatformC.h
  - SPINNAKERC\_API, [464](#)
- spinNodeCallbackFunction
  - SpinnakerGenApiDefsC.h, [456](#)
- spinNodeCallbackHandle
  - SpinnakerGenApiDefsC.h, [456](#)
- spinNodeDeregisterCallback
  - SpinnakerGenApiC.h, [433](#)
- spinNodeFromString
  - SpinnakerGenApiC.h, [434](#)
- spinNodeFromStringEx
  - SpinnakerGenApiC.h, [434](#)
- spinNodeGetAccessMode
  - SpinnakerGenApiC.h, [435](#)
- spinNodeGetCachingMode
  - SpinnakerGenApiC.h, [435](#)
- spinNodeGetDescription
  - SpinnakerGenApiC.h, [436](#)
- spinNodeGetDisplayName
  - SpinnakerGenApiC.h, [436](#)
- spinNodeGetImposedAccessMode
  - SpinnakerGenApiC.h, [437](#)
- spinNodeGetImposedVisibility
  - SpinnakerGenApiC.h, [437](#)
- spinNodeGetName
  - SpinnakerGenApiC.h, [438](#)
- spinNodeGetNameSpace
  - SpinnakerGenApiC.h, [438](#)
- spinNodeGetPollingTime
  - SpinnakerGenApiC.h, [439](#)
- spinNodeGetToolTip
  - SpinnakerGenApiC.h, [439](#)
- spinNodeGetType
  - SpinnakerGenApiC.h, [440](#)
- spinNodeGetVisibility
  - SpinnakerGenApiC.h, [440](#)
- spinNodeHandle
  - SpinnakerGenApiDefsC.h, [456](#)
- spinNodeInvalidateNode
  - SpinnakerGenApiC.h, [441](#)
- spinNodesAvailable
  - SpinnakerGenApiC.h, [441](#)
- spinNodesEqual
  - SpinnakerGenApiC.h, [442](#)
- spinNodesImplemented
  - SpinnakerGenApiC.h, [442](#)
- spinNodesReadable
  - SpinnakerGenApiC.h, [443](#)
- spinNodesWritable
  - SpinnakerGenApiC.h, [443](#)
- spinNodeMapGetNode
  - SpinnakerGenApiC.h, [444](#)
- spinNodeMapGetNodeByIndex
  - SpinnakerGenApiC.h, [444](#)
- spinNodeMapGetNumNodes
  - SpinnakerGenApiC.h, [445](#)
- spinNodeMapHandle
  - SpinnakerGenApiDefsC.h, [456](#)
- spinNodeMapPoll
  - SpinnakerGenApiC.h, [445](#)
- spinNodeRegisterCallback
  - SpinnakerGenApiC.h, [446](#)
- spinNodeToString
  - SpinnakerGenApiC.h, [446](#)
- spinNodeToStringEx
  - SpinnakerGenApiC.h, [447](#)
- spinRegisterGet
  - SpinnakerGenApiC.h, [447](#)
- spinRegisterGetAddress
  - SpinnakerGenApiC.h, [448](#)
- spinRegisterGetEx
  - SpinnakerGenApiC.h, [448](#)
- spinRegisterGetLength
  - SpinnakerGenApiC.h, [449](#)
- spinRegisterSet
  - SpinnakerGenApiC.h, [449](#)
- spinRegisterSetEx
  - SpinnakerGenApiC.h, [450](#)
- spinRegisterSetReference



- SpinnakerGenApiC.h, [450](#)
- spinRemovalEventFunction
  - SpinnakerDefsC.h, [407](#)
- spinStringGetMaxLength
  - SpinnakerGenApiC.h, [451](#)
- spinStringGetValue
  - SpinnakerGenApiC.h, [451](#)
- spinStringGetValueEx
  - SpinnakerGenApiC.h, [452](#)
- spinStringSetValue
  - SpinnakerGenApiC.h, [452](#)
- spinStringSetValueEx
  - SpinnakerGenApiC.h, [453](#)
- spinSystem
  - SpinnakerDefsC.h, [408](#)
- spinSystemGetCameras
  - SpinnakerC.h, [389](#)
- spinSystemGetCamerasEx
  - SpinnakerC.h, [389](#)
- spinSystemGetInstance
  - SpinnakerC.h, [390](#)
- spinSystemGetInterfaces
  - SpinnakerC.h, [390](#)
- spinSystemGetLibraryVersion
  - SpinnakerC.h, [391](#)
- spinSystemGetLoggingLevel
  - SpinnakerC.h, [391](#)
- spinSystemGetTLNodeMap
  - SpinnakerC.h, [391](#)
- spinSystemIsInUse
  - SpinnakerC.h, [392](#)
- spinSystemRegisterDeviceArrivalEventHandler
  - SpinnakerC.h, [392](#)
- spinSystemRegisterDeviceRemovalEventHandler
  - SpinnakerC.h, [393](#)
- spinSystemRegisterInterfaceEventHandler
  - SpinnakerC.h, [393](#)
- spinSystemRegisterLogEventHandler
  - SpinnakerC.h, [394](#)
- spinSystemReleaseInstance
  - SpinnakerC.h, [394](#)
- spinSystemSendActionCommand
  - SpinnakerC.h, [395](#)
- spinSystemSetLoggingLevel
  - SpinnakerC.h, [396](#)
- spinSystemUnregisterAllLogEventHandlers
  - SpinnakerC.h, [396](#)
- spinSystemUnregisterDeviceArrivalEventHandler
  - SpinnakerC.h, [396](#)
- spinSystemUnregisterDeviceRemovalEventHandler
  - SpinnakerC.h, [397](#)
- spinSystemUnregisterInterfaceEventHandler
  - SpinnakerC.h, [397](#)
- spinSystemUnregisterLogEventHandler
  - SpinnakerC.h, [398](#)
- spinSystemUpdateCameras
  - SpinnakerC.h, [398](#)
- spinSystemUpdateCamerasEx
  - SpinnakerC.h, [399](#)
- spinVideo
  - SpinnakerDefsC.h, [408](#)
- SpinVideo Recording Access, [44](#)
- spinVideoAppend
  - SpinVideoC.h, [465](#)
- SpinVideoC.h
  - spinVideoAppend, [465](#)
  - spinVideoClose, [466](#)
  - spinVideoOpenH264, [466](#)
  - spinVideoOpenMJPEG, [466](#)
  - spinVideoOpenUncompressed, [466](#)
  - spinVideoSetMaximumFileSize, [466](#)
- spinVideoClose
  - SpinVideoC.h, [466](#)
- spinVideoOpenH264
  - SpinVideoC.h, [466](#)
- spinVideoOpenMJPEG
  - SpinVideoC.h, [466](#)
- spinVideoOpenUncompressed
  - SpinVideoC.h, [466](#)
- spinVideoSetMaximumFileSize
  - SpinVideoC.h, [466](#)
- Standard
  - SpinnakerGenApiDefsC.h, [460](#)
- Status
  - \_actionCommandResult, [51](#)
- StreamAnnounceBufferMinimum
  - \_quickSpinTLStream, [163](#)
- StreamAnnouncedBufferCount
  - \_quickSpinTLStream, [163](#)
- StreamBlockTransferSize
  - \_quickSpinTLStream, [163](#)
- StreamBufferAlignment
  - \_quickSpinTLStream, [163](#)
- StreamBufferCountManual
  - \_quickSpinTLStream, [164](#)
- StreamBufferCountMax
  - \_quickSpinTLStream, [164](#)
- StreamBufferCountMode
  - \_quickSpinTLStream, [164](#)
- StreamBufferCountMode\_Auto
  - TransportLayerDefsC.h, [473](#)
- StreamBufferCountMode\_Manual
  - TransportLayerDefsC.h, [473](#)
- StreamBufferCountResult
  - \_quickSpinTLStream, [164](#)
- StreamBufferHandlingMode
  - \_quickSpinTLStream, [164](#)
- StreamBufferHandlingMode\_NewestFirst
  - TransportLayerDefsC.h, [474](#)
- StreamBufferHandlingMode\_NewestOnly
  - TransportLayerDefsC.h, [474](#)
- StreamBufferHandlingMode\_OldestFirst
  - TransportLayerDefsC.h, [474](#)
- StreamBufferHandlingMode\_OldestFirstOverwrite
  - TransportLayerDefsC.h, [474](#)
- StreamChunkCountMaximum

- [\\_quickSpinTLStream](#), 164
- [StreamCRCCheckEnable](#)
  - [\\_quickSpinTLStream](#), 164
- [StreamDeliveredFrameCount](#)
  - [\\_quickSpinTLStream](#), 164
- [StreamFailedBufferCount](#)
  - [\\_quickSpinTLStream](#), 165
- [StreamID](#)
  - [\\_quickSpinTLStream](#), 165
- [StreamInputBufferCount](#)
  - [\\_quickSpinTLStream](#), 165
- [StreamIsGrabbing](#)
  - [\\_quickSpinTLStream](#), 165
- [StreamLostFrameCount](#)
  - [\\_quickSpinTLStream](#), 165
- [StreamOutputBufferCount](#)
  - [\\_quickSpinTLStream](#), 165
- [StreamStartedFrameCount](#)
  - [\\_quickSpinTLStream](#), 165
- [StreamType](#)
  - [\\_quickSpinTLStream](#), 165
- [StreamType\\_CameraLink](#)
  - [TransportLayerDefsC.h](#), 474
- [StreamType\\_CameraLinkHS](#)
  - [TransportLayerDefsC.h](#), 474
- [StreamType\\_CoaxPress](#)
  - [TransportLayerDefsC.h](#), 474
- [StreamType\\_Custom](#)
  - [TransportLayerDefsC.h](#), 474
- [StreamType\\_GigEVision](#)
  - [TransportLayerDefsC.h](#), 474
- [StreamType\\_USB3Vision](#)
  - [TransportLayerDefsC.h](#), 474
- [String Access](#), 33
- [StringNode](#)
  - [SpinnakerGenApiDefsC.h](#), 461
- [System Access](#), 14
- [Test0001](#)
  - [\\_quickSpin](#), 140
- [TestEventGenerate](#)
  - [\\_quickSpin](#), 140
- [TestPattern](#)
  - [\\_quickSpin](#), 140
- [TestPattern\\_Increment](#)
  - [CameraDefsC.h](#), 292
- [TestPattern\\_Off](#)
  - [CameraDefsC.h](#), 292
- [TestPattern\\_SensorTestPattern](#)
  - [CameraDefsC.h](#), 292
- [TestPatternGeneratorSelector](#)
  - [\\_quickSpin](#), 140
- [TestPatternGeneratorSelector\\_PipelineStart](#)
  - [CameraDefsC.h](#), 292
- [TestPatternGeneratorSelector\\_Sensor](#)
  - [CameraDefsC.h](#), 292
- [TestPendingAck](#)
  - [\\_quickSpin](#), 140
- [TIFF](#)
  - [SpinnakerDefsC.h](#), 410
- [TimerDelay](#)
  - [\\_quickSpin](#), 140
- [TimerDuration](#)
  - [\\_quickSpin](#), 141
- [TimerReset](#)
  - [\\_quickSpin](#), 141
- [TimerSelector](#)
  - [\\_quickSpin](#), 141
- [TimerSelector\\_Timer0](#)
  - [CameraDefsC.h](#), 293
- [TimerSelector\\_Timer1](#)
  - [CameraDefsC.h](#), 293
- [TimerSelector\\_Timer2](#)
  - [CameraDefsC.h](#), 293
- [TimerStatus](#)
  - [\\_quickSpin](#), 141
- [TimerStatus\\_TimerActive](#)
  - [CameraDefsC.h](#), 293
- [TimerStatus\\_TimerCompleted](#)
  - [CameraDefsC.h](#), 293
- [TimerStatus\\_TimerIdle](#)
  - [CameraDefsC.h](#), 293
- [TimerStatus\\_TimerTriggerWait](#)
  - [CameraDefsC.h](#), 293
- [TimerTriggerActivation](#)
  - [\\_quickSpin](#), 141
- [TimerTriggerActivation\\_AnyEdge](#)
  - [CameraDefsC.h](#), 293
- [TimerTriggerActivation\\_FallingEdge](#)
  - [CameraDefsC.h](#), 293
- [TimerTriggerActivation\\_LevelHigh](#)
  - [CameraDefsC.h](#), 293
- [TimerTriggerActivation\\_LevelLow](#)
  - [CameraDefsC.h](#), 293
- [TimerTriggerActivation\\_RisingEdge](#)
  - [CameraDefsC.h](#), 293
- [TimerTriggerSource](#)
  - [\\_quickSpin](#), 141
- [TimerTriggerSource\\_AcquisitionEnd](#)
  - [CameraDefsC.h](#), 294
- [TimerTriggerSource\\_AcquisitionStart](#)
  - [CameraDefsC.h](#), 294
- [TimerTriggerSource\\_AcquisitionTrigger](#)
  - [CameraDefsC.h](#), 294
- [TimerTriggerSource\\_Action0](#)
  - [CameraDefsC.h](#), 295
- [TimerTriggerSource\\_Action1](#)
  - [CameraDefsC.h](#), 295
- [TimerTriggerSource\\_Action2](#)
  - [CameraDefsC.h](#), 295
- [TimerTriggerSource\\_Counter0End](#)
  - [CameraDefsC.h](#), 294
- [TimerTriggerSource\\_Counter0Start](#)
  - [CameraDefsC.h](#), 294
- [TimerTriggerSource\\_Counter1End](#)
  - [CameraDefsC.h](#), 294
- [TimerTriggerSource\\_Counter1Start](#)

- CameraDefsC.h, [294](#)
- TimerTriggerSource\_Counter2End
  - CameraDefsC.h, [294](#)
- TimerTriggerSource\_Counter2Start
  - CameraDefsC.h, [294](#)
- TimerTriggerSource\_Encoder0
  - CameraDefsC.h, [295](#)
- TimerTriggerSource\_Encoder1
  - CameraDefsC.h, [295](#)
- TimerTriggerSource\_Encoder2
  - CameraDefsC.h, [295](#)
- TimerTriggerSource\_ExposureEnd
  - CameraDefsC.h, [294](#)
- TimerTriggerSource\_ExposureStart
  - CameraDefsC.h, [294](#)
- TimerTriggerSource\_FrameBurstEnd
  - CameraDefsC.h, [294](#)
- TimerTriggerSource\_FrameBurstStart
  - CameraDefsC.h, [294](#)
- TimerTriggerSource\_FrameEnd
  - CameraDefsC.h, [294](#)
- TimerTriggerSource\_FrameStart
  - CameraDefsC.h, [294](#)
- TimerTriggerSource\_FrameTrigger
  - CameraDefsC.h, [294](#)
- TimerTriggerSource\_Line0
  - CameraDefsC.h, [294](#)
- TimerTriggerSource\_Line1
  - CameraDefsC.h, [294](#)
- TimerTriggerSource\_Line2
  - CameraDefsC.h, [294](#)
- TimerTriggerSource\_LineEnd
  - CameraDefsC.h, [294](#)
- TimerTriggerSource\_LineStart
  - CameraDefsC.h, [294](#)
- TimerTriggerSource\_LineTrigger
  - CameraDefsC.h, [294](#)
- TimerTriggerSource\_LinkTrigger0
  - CameraDefsC.h, [295](#)
- TimerTriggerSource\_LinkTrigger1
  - CameraDefsC.h, [295](#)
- TimerTriggerSource\_LinkTrigger2
  - CameraDefsC.h, [295](#)
- TimerTriggerSource\_Off
  - CameraDefsC.h, [294](#)
- TimerTriggerSource\_SoftwareSignal0
  - CameraDefsC.h, [295](#)
- TimerTriggerSource\_SoftwareSignal1
  - CameraDefsC.h, [295](#)
- TimerTriggerSource\_SoftwareSignal2
  - CameraDefsC.h, [295](#)
- TimerTriggerSource\_Timer0End
  - CameraDefsC.h, [294](#)
- TimerTriggerSource\_Timer0Start
  - CameraDefsC.h, [294](#)
- TimerTriggerSource\_Timer1End
  - CameraDefsC.h, [294](#)
- TimerTriggerSource\_Timer1Start
  - CameraDefsC.h, [294](#)
- TimerTriggerSource\_Timer2End
  - CameraDefsC.h, [294](#)
- TimerTriggerSource\_Timer2Start
  - CameraDefsC.h, [294](#)
- TimerTriggerSource\_UserOutput0
  - CameraDefsC.h, [294](#)
- TimerTriggerSource\_UserOutput1
  - CameraDefsC.h, [294](#)
- TimerTriggerSource\_UserOutput2
  - CameraDefsC.h, [294](#)
- TimerValue
  - \_quickSpin, [141](#)
- Timestamp
  - \_quickSpin, [141](#)
- TimestampLatch
  - \_quickSpin, [142](#)
- TimestampLatchValue
  - \_quickSpin, [142](#)
- TimestampReset
  - \_quickSpin, [142](#)
- TLDevice Structures, [46](#)
- TLDisplayName
  - \_quickSpinTLSystem, [168](#)
- TLFileName
  - \_quickSpinTLSystem, [168](#)
- TLID
  - \_quickSpinTLSystem, [169](#)
- TLInterface Structures, [47](#)
- TLModelName
  - \_quickSpinTLSystem, [169](#)
- TLParamsLocked
  - \_quickSpin, [142](#)
- TLPath
  - \_quickSpinTLSystem, [169](#)
- TLStream Structures, [48](#)
- TLSystem Structures, [49](#)
- TLType
  - \_quickSpinTLSystem, [169](#)
- TLType\_CameraLink
  - TransportLayerDefsC.h, [475](#)
- TLType\_CameraLinkHS
  - TransportLayerDefsC.h, [475](#)
- TLType\_CoaXPress
  - TransportLayerDefsC.h, [475](#)
- TLType\_Custom
  - TransportLayerDefsC.h, [475](#)
- TLType\_GigEVision
  - TransportLayerDefsC.h, [475](#)
- TLType\_Mixed
  - TransportLayerDefsC.h, [475](#)
- TLType\_USB3Vision
  - TransportLayerDefsC.h, [475](#)
- TLVendorName
  - \_quickSpinTLSystem, [169](#)
- TLVersion
  - \_quickSpinTLSystem, [169](#)
- TransferAbort

- [\\_quickSpin](#), 142
- TransferBlockCount
  - [\\_quickSpin](#), 142
- TransferBurstCount
  - [\\_quickSpin](#), 142
- TransferComponentSelector
  - [\\_quickSpin](#), 142
- TransferComponentSelector\_All
  - [CameraDefsC.h](#), 295
- TransferComponentSelector\_Blue
  - [CameraDefsC.h](#), 295
- TransferComponentSelector\_Green
  - [CameraDefsC.h](#), 295
- TransferComponentSelector\_Red
  - [CameraDefsC.h](#), 295
- TransferControlMode
  - [\\_quickSpin](#), 143
- TransferControlMode\_Automatic
  - [CameraDefsC.h](#), 296
- TransferControlMode\_Basic
  - [CameraDefsC.h](#), 296
- TransferControlMode\_UserControlled
  - [CameraDefsC.h](#), 296
- TransferOperationMode
  - [\\_quickSpin](#), 143
- TransferOperationMode\_Continuous
  - [CameraDefsC.h](#), 296
- TransferOperationMode\_MultiBlock
  - [CameraDefsC.h](#), 296
- TransferPause
  - [\\_quickSpin](#), 143
- TransferQueueCurrentBlockCount
  - [\\_quickSpin](#), 143
- TransferQueueMaxBlockCount
  - [\\_quickSpin](#), 143
- TransferQueueMode
  - [\\_quickSpin](#), 143
- TransferQueueMode\_FirstInFirstOut
  - [CameraDefsC.h](#), 296
- TransferQueueOverflowCount
  - [\\_quickSpin](#), 143
- TransferResume
  - [\\_quickSpin](#), 143
- TransferSelector
  - [\\_quickSpin](#), 144
- TransferSelector\_All
  - [CameraDefsC.h](#), 296
- TransferSelector\_Stream0
  - [CameraDefsC.h](#), 296
- TransferSelector\_Stream1
  - [CameraDefsC.h](#), 296
- TransferSelector\_Stream2
  - [CameraDefsC.h](#), 296
- TransferStart
  - [\\_quickSpin](#), 144
- TransferStatus
  - [\\_quickSpin](#), 144
- TransferStatusSelector
  - [\\_quickSpin](#), 144
- TransferStatusSelector\_Paused
  - [CameraDefsC.h](#), 297
- TransferStatusSelector\_QueueOverflow
  - [CameraDefsC.h](#), 297
- TransferStatusSelector\_Stopped
  - [CameraDefsC.h](#), 297
- TransferStatusSelector\_Stopping
  - [CameraDefsC.h](#), 297
- TransferStatusSelector\_Streaming
  - [CameraDefsC.h](#), 297
- TransferStop
  - [\\_quickSpin](#), 144
- TransferStreamChannel
  - [\\_quickSpin](#), 144
- TransferTriggerActivation
  - [\\_quickSpin](#), 144
- TransferTriggerActivation\_AnyEdge
  - [CameraDefsC.h](#), 297
- TransferTriggerActivation\_FallingEdge
  - [CameraDefsC.h](#), 297
- TransferTriggerActivation\_LevelHigh
  - [CameraDefsC.h](#), 297
- TransferTriggerActivation\_LevelLow
  - [CameraDefsC.h](#), 297
- TransferTriggerActivation\_RisingEdge
  - [CameraDefsC.h](#), 297
- TransferTriggerMode
  - [\\_quickSpin](#), 144
- TransferTriggerMode\_Off
  - [CameraDefsC.h](#), 298
- TransferTriggerMode\_On
  - [CameraDefsC.h](#), 298
- TransferTriggerSelector
  - [\\_quickSpin](#), 145
- TransferTriggerSelector\_TransferAbort
  - [CameraDefsC.h](#), 298
- TransferTriggerSelector\_TransferActive
  - [CameraDefsC.h](#), 298
- TransferTriggerSelector\_TransferBurstStart
  - [CameraDefsC.h](#), 298
- TransferTriggerSelector\_TransferBurstStop
  - [CameraDefsC.h](#), 298
- TransferTriggerSelector\_TransferPause
  - [CameraDefsC.h](#), 298
- TransferTriggerSelector\_TransferResume
  - [CameraDefsC.h](#), 298
- TransferTriggerSelector\_TransferStart
  - [CameraDefsC.h](#), 298
- TransferTriggerSelector\_TransferStop
  - [CameraDefsC.h](#), 298
- TransferTriggerSource
  - [\\_quickSpin](#), 145
- TransferTriggerSource\_Action0
  - [CameraDefsC.h](#), 299
- TransferTriggerSource\_Action1
  - [CameraDefsC.h](#), 299
- TransferTriggerSource\_Action2

- CameraDefsC.h, [299](#)
- TransferTriggerSource\_Counter0End
  - CameraDefsC.h, [299](#)
- TransferTriggerSource\_Counter0Start
  - CameraDefsC.h, [298](#)
- TransferTriggerSource\_Counter1End
  - CameraDefsC.h, [299](#)
- TransferTriggerSource\_Counter1Start
  - CameraDefsC.h, [298](#)
- TransferTriggerSource\_Counter2End
  - CameraDefsC.h, [299](#)
- TransferTriggerSource\_Counter2Start
  - CameraDefsC.h, [299](#)
- TransferTriggerSource\_Line0
  - CameraDefsC.h, [298](#)
- TransferTriggerSource\_Line1
  - CameraDefsC.h, [298](#)
- TransferTriggerSource\_Line2
  - CameraDefsC.h, [298](#)
- TransferTriggerSource\_SoftwareSignal0
  - CameraDefsC.h, [299](#)
- TransferTriggerSource\_SoftwareSignal1
  - CameraDefsC.h, [299](#)
- TransferTriggerSource\_SoftwareSignal2
  - CameraDefsC.h, [299](#)
- TransferTriggerSource\_Timer0End
  - CameraDefsC.h, [299](#)
- TransferTriggerSource\_Timer0Start
  - CameraDefsC.h, [299](#)
- TransferTriggerSource\_Timer1End
  - CameraDefsC.h, [299](#)
- TransferTriggerSource\_Timer1Start
  - CameraDefsC.h, [299](#)
- TransferTriggerSource\_Timer2End
  - CameraDefsC.h, [299](#)
- TransferTriggerSource\_Timer2Start
  - CameraDefsC.h, [299](#)
- Transport Layer Enumerations, [45](#)
- TransportLayerDefsC.h
  - \_spinTLDeviceAccessStatusEnums, [469](#)
  - \_spinTLDeviceCurrentSpeedEnums, [469](#)
  - \_spinTLDeviceEndiannessMechanismEnums, [469](#)
  - \_spinTLDeviceTypeEnums, [471](#)
  - \_spinTLFilterDriverStatusEnums, [471](#)
  - \_spinTLGUIXMLLocationEnums, [472](#)
  - \_spinTLGenICamXMLLocationEnums, [471](#)
  - \_spinTLGevCCPEnums, [472](#)
  - \_spinTLInterfaceTypeEnums, [472](#)
  - \_spinTLPOEStatusEnums, [473](#)
  - \_spinTLStreamBufferCountModeEnums, [473](#)
  - \_spinTLStreamBufferHandlingModeEnums, [473](#)
  - \_spinTLStreamTypeEnums, [474](#)
  - \_spinTLTLTypeEnums, [474](#)
  - DeviceAccessStatus\_Busy, [469](#)
  - DeviceAccessStatus\_NoAccess, [469](#)
  - DeviceAccessStatus\_OpenReadOnly, [469](#)
  - DeviceAccessStatus\_OpenReadWrite, [469](#)
  - DeviceAccessStatus\_ReadOnly, [469](#)
  - DeviceAccessStatus\_ReadWrite, [469](#)
  - DeviceAccessStatus\_Unknown, [469](#)
  - DeviceCurrentSpeed\_FullSpeed, [469](#)
  - DeviceCurrentSpeed\_HighSpeed, [469](#)
  - DeviceCurrentSpeed\_LowSpeed, [469](#)
  - DeviceCurrentSpeed\_SuperSpeed, [469](#)
  - DeviceCurrentSpeed\_UnknownSpeed, [469](#)
  - DeviceEndiannessMechanism\_Legacy, [471](#)
  - DeviceEndiannessMechanism\_Standard, [471](#)
  - DeviceType\_CameraLink, [471](#)
  - DeviceType\_CameraLinkHS, [471](#)
  - DeviceType\_CoaXPress, [471](#)
  - DeviceType\_Custom, [471](#)
  - DeviceType\_GigEVision, [471](#)
  - DeviceType\_USB3Vision, [471](#)
  - FilterDriverStatus\_Disabled, [471](#)
  - FilterDriverStatus\_Enabled, [471](#)
  - FilterDriverStatus\_NotSupported, [471](#)
  - GenICamXMLLocation\_Device, [472](#)
  - GenICamXMLLocation\_Host, [472](#)
  - GevCCP\_EnumEntry\_GevCCP\_ControlAccess, [472](#)
  - GevCCP\_EnumEntry\_GevCCP\_ExclusiveAccess, [472](#)
  - GevCCP\_EnumEntry\_GevCCP\_OpenAccess, [472](#)
  - GUIXMLLocation\_Device, [472](#)
  - GUIXMLLocation\_Host, [472](#)
  - InterfaceType\_CameraLink, [472](#)
  - InterfaceType\_CameraLinkHS, [473](#)
  - InterfaceType\_CoaXPress, [473](#)
  - InterfaceType\_Custom, [473](#)
  - InterfaceType\_GigEVision, [472](#)
  - InterfaceType\_USB3Vision, [473](#)
  - NUMDEVICEACCESSSTATUS, [469](#)
  - NUMDEVICECURRENTSPEED, [469](#)
  - NUMDEVICEENDIANESSMECHANISM, [471](#)
  - NUMDEVICETYPE, [471](#)
  - NUMFILTERDRIVERSTATUS, [471](#)
  - NUMGENICAMXMLLOCATION, [472](#)
  - NUMGEVCCP, [472](#)
  - NUMGUIXMLLOCATION, [472](#)
  - NUMINTERFACETYPE, [473](#)
  - NUMPOESTATUS, [473](#)
  - NUMSTREAMBUFFERCOUNTMODE, [473](#)
  - NUMSTREAMBUFFERHANDLINGMODE, [474](#)
  - NUMSTREAMTYPE, [474](#)
  - NUMTLTYPE, [475](#)
  - POEStatus\_NotSupported, [473](#)
  - POEStatus\_PowerOff, [473](#)
  - POEStatus\_PowerOn, [473](#)
  - StreamBufferCountMode\_Auto, [473](#)
  - StreamBufferCountMode\_Manual, [473](#)
  - StreamBufferHandlingMode\_NewestFirst, [474](#)
  - StreamBufferHandlingMode\_NewestOnly, [474](#)
  - StreamBufferHandlingMode\_OldestFirst, [474](#)
  - StreamBufferHandlingMode\_OldestFirstOverwrite, [474](#)
  - StreamType\_CameraLink, [474](#)

- StreamType\_CameraLinkHS, [474](#)
- StreamType\_CoaXPress, [474](#)
- StreamType\_Custom, [474](#)
- StreamType\_GigEVision, [474](#)
- StreamType\_USB3Vision, [474](#)
- TLType\_CameraLink, [475](#)
- TLType\_CameraLinkHS, [475](#)
- TLType\_CoaXPress, [475](#)
- TLType\_Custom, [475](#)
- TLType\_GigEVision, [475](#)
- TLType\_Mixed, [475](#)
- TLType\_USB3Vision, [475](#)
- TriggerActivation
  - \_quickSpin, [145](#)
- TriggerActivation\_AnyEdge
  - CameraDefsC.h, [299](#)
- TriggerActivation\_FallingEdge
  - CameraDefsC.h, [299](#)
- TriggerActivation\_LevelHigh
  - CameraDefsC.h, [299](#)
- TriggerActivation\_LevelLow
  - CameraDefsC.h, [299](#)
- TriggerActivation\_RisingEdge
  - CameraDefsC.h, [299](#)
- TriggerDelay
  - \_quickSpin, [145](#)
- TriggerDivider
  - \_quickSpin, [145](#)
- TriggerEventTest
  - \_quickSpin, [145](#)
- TriggerMode
  - \_quickSpin, [145](#)
- TriggerMode\_Off
  - CameraDefsC.h, [300](#)
- TriggerMode\_On
  - CameraDefsC.h, [300](#)
- TriggerMultiplier
  - \_quickSpin, [145](#)
- TriggerOverlap
  - \_quickSpin, [146](#)
- TriggerOverlap\_Off
  - CameraDefsC.h, [300](#)
- TriggerOverlap\_PreviousFrame
  - CameraDefsC.h, [300](#)
- TriggerOverlap\_ReadOut
  - CameraDefsC.h, [300](#)
- TriggerSelector
  - \_quickSpin, [146](#)
- TriggerSelector\_AcquisitionStart
  - CameraDefsC.h, [300](#)
- TriggerSelector\_FrameBurstStart
  - CameraDefsC.h, [300](#)
- TriggerSelector\_FrameStart
  - CameraDefsC.h, [300](#)
- TriggerSoftware
  - \_quickSpin, [146](#)
- TriggerSource
  - \_quickSpin, [146](#)
- TriggerSource\_Action0
  - CameraDefsC.h, [301](#)
- TriggerSource\_Counter0End
  - CameraDefsC.h, [301](#)
- TriggerSource\_Counter0Start
  - CameraDefsC.h, [301](#)
- TriggerSource\_Counter1End
  - CameraDefsC.h, [301](#)
- TriggerSource\_Counter1Start
  - CameraDefsC.h, [301](#)
- TriggerSource\_Line0
  - CameraDefsC.h, [301](#)
- TriggerSource\_Line1
  - CameraDefsC.h, [301](#)
- TriggerSource\_Line2
  - CameraDefsC.h, [301](#)
- TriggerSource\_Line3
  - CameraDefsC.h, [301](#)
- TriggerSource\_LogicBlock0
  - CameraDefsC.h, [301](#)
- TriggerSource\_LogicBlock1
  - CameraDefsC.h, [301](#)
- TriggerSource\_Software
  - CameraDefsC.h, [301](#)
- TriggerSource\_UserOutput0
  - CameraDefsC.h, [301](#)
- TriggerSource\_UserOutput1
  - CameraDefsC.h, [301](#)
- TriggerSource\_UserOutput2
  - CameraDefsC.h, [301](#)
- TriggerSource\_UserOutput3
  - CameraDefsC.h, [301](#)
- True
  - SpinnakerDefsC.h, [414](#)
- type
  - \_spinLibraryVersion, [181](#)
- UNKNOWN\_PIXELFORMAT
  - CameraDefsC.h, [275](#)
- UnknownNode
  - SpinnakerGenApiDefsC.h, [461](#)
- Unsigned
  - SpinnakerGenApiDefsC.h, [462](#)
- USB
  - SpinnakerGenApiDefsC.h, [462](#)
- UserOutputSelector
  - \_quickSpin, [146](#)
- UserOutputSelector\_UserOutput0
  - CameraDefsC.h, [301](#)
- UserOutputSelector\_UserOutput1
  - CameraDefsC.h, [301](#)
- UserOutputSelector\_UserOutput2
  - CameraDefsC.h, [301](#)
- UserOutputSelector\_UserOutput3
  - CameraDefsC.h, [301](#)
- UserOutputValue
  - \_quickSpin, [146](#)
- UserOutputValueAll
  - \_quickSpin, [146](#)



- UserOutputValueAllMask
  - \_quickSpin, [146](#)
- UserSetDefault
  - \_quickSpin, [147](#)
- UserSetDefault\_Default
  - CameraDefsC.h, [302](#)
- UserSetDefault\_UserSet0
  - CameraDefsC.h, [302](#)
- UserSetDefault\_UserSet1
  - CameraDefsC.h, [302](#)
- UserSetFeatureEnable
  - \_quickSpin, [147](#)
- UserSetLoad
  - \_quickSpin, [147](#)
- UserSetSave
  - \_quickSpin, [147](#)
- UserSetSelector
  - \_quickSpin, [147](#)
- UserSetSelector\_Default
  - CameraDefsC.h, [302](#)
- UserSetSelector\_UserSet0
  - CameraDefsC.h, [302](#)
- UserSetSelector\_UserSet1
  - CameraDefsC.h, [302](#)
  
- V3\_3Enable
  - \_quickSpin, [147](#)
- ValueNode
  - SpinnakerGenApiDefsC.h, [461](#)
- Varying
  - SpinnakerGenApiDefsC.h, [462](#)
  
- WEIGHTED\_DIRECTIONAL\_FILTER
  - SpinnakerDefsC.h, [409](#)
- WhiteClip
  - \_quickSpin, [147](#)
- WhiteClipSelector
  - \_quickSpin, [147](#)
- WhiteClipSelector\_All
  - CameraDefsC.h, [302](#)
- WhiteClipSelector\_Blue
  - CameraDefsC.h, [302](#)
- WhiteClipSelector\_Green
  - CameraDefsC.h, [302](#)
- WhiteClipSelector\_Red
  - CameraDefsC.h, [302](#)
- WhiteClipSelector\_Tap1
  - CameraDefsC.h, [302](#)
- WhiteClipSelector\_Tap2
  - CameraDefsC.h, [302](#)
- WhiteClipSelector\_U
  - CameraDefsC.h, [302](#)
- WhiteClipSelector\_V
  - CameraDefsC.h, [302](#)
- WhiteClipSelector\_Y
  - CameraDefsC.h, [302](#)
- Width
  - \_quickSpin, [148](#)
- width
  - \_spinH264Option, [178](#)
- WidthMax
  - \_quickSpin, [148](#)
- WO
  - SpinnakerGenApiDefsC.h, [457](#)
- WriteAround
  - SpinnakerGenApiDefsC.h, [457](#)
- WriteThrough
  - SpinnakerGenApiDefsC.h, [457](#)
  
- xvAll
  - SpinnakerGenApiDefsC.h, [463](#)
- xvCycles
  - SpinnakerGenApiDefsC.h, [463](#)
- xvDefault
  - SpinnakerGenApiDefsC.h, [463](#)
- xvLoad
  - SpinnakerGenApiDefsC.h, [463](#)
- xvSFNC
  - SpinnakerGenApiDefsC.h, [463](#)
  
- Yes
  - SpinnakerGenApiDefsC.h, [464](#)